Monadic Second-Order Logic
with Arbitrary Monadic Predicates

NATHANAËL FIJALKOW, University of Wariwck

CHARLES PAPERMAN, Institut Mathématiques de Jussieu - Paris rive gauche

1 INTRODUCTION

Monadic Second-Order Logic (**MSO**) over finite words equipped with the linear ordering on positions is a well-studied and understood logic. It provides a mathematical framework for applications in many areas such as program verification, database theory and linguistics. In 1962, Büchi [Büc62] proved the decidability of the satisfiability problem for **MSO** formulae.

1.1 Uniform Monadic Predicates

In 1966, Elgot and Rabin [ER66] considered extensions of **MSO** with uniform monadic predicates. For instance, the following formula

$$\forall x, \ a(x) \iff x \text{ is prime },$$

describes the set of finite words such that the letter a appears exactly in prime positions. The predicate "x is a prime number" is a *uniform numerical monadic* predicate. Being *numerical* means that its interpretation only depends on positions, *i.e.* $\mathbf{P} = (\mathbf{P}_n)_{n \in \mathbb{N}}$, *uniform* means that it can be seen as a relation over integers, *i.e.* $\mathbf{P} \subseteq \mathbb{N}^k$, and *monadic* means that it has arity 1, *i.e.* $k = 1$. Elgot and Rabin were interested in the following question: for a uniform numerical monadic predicate $\mathbf{P} \subseteq \mathbb{N}$, is the satisfiability problem of $\mathbf{MSO}[\leq, \mathbf{P}]$ decidable? A series of papers gave tighter conditions on $\mathbf{P}$, culminating to two final answers: in 1984, Semenov [Sem84] gave a characterization of the predicates $\mathbf{P}$ such that $\mathbf{MSO}[\leq, \mathbf{P}]$ is decidable, and in 2006, Rabinovich and Thomas [Rab07, RT06] proved this characterization to be equivalent to the predicate $\mathbf{P}$ being effectively profinitely ultimately periodic. Further questions on uniform monadic predicates have been investigated. For instance,

39

Rabinovich [Rab12] gave a solution to the Church synthesis problem for $\mathbf{MSO}[\leq, \mathbf{P}]$, for a large class of predicates $\mathbf{P}$.

In this paper, we consider the so-called numerical monadic predicates and not only the uniform ones: such a predicate $\mathbf{P}$ is given, for each length $n \in \mathbb{N}$, by a predicate over the first n positions $\mathbf{P}_n \subseteq \{0, \ldots, n-1\}$. The set Arb_1 of these predicates contains the set $\mathrm{Arb}_1^{\mathrm{u}}$ of uniform monadic predicates. Note that the subscript 1 in Arb_1 and $\mathrm{Arb}_1^{\mathrm{u}}$ corresponds to the arity. A formal definition can be found in Section 2.1.

1.2 Advice Regular Languages

We call languages definable in $\mathbf{MSO}[\leq, \mathrm{Arb}_1]$ *advice regular*. Note that no computability assumptions are made on the monadic predicates, so this class contains undecidable languages. Our first contribution is to give equivalent presentations of this class, which is a Boolean algebra extending the class of regular languages:

(1) It has an equivalent automaton model: *automata with advice*.
(2) It has an equivalent algebraic model: *one-scan programs*.
(3) It has a machine-independent characterization, based on generalizations of Myhill-Nerode equivalence relations.

This extends the equivalence between automata with advice and Myhill-Nerode equivalence relations proved in [KRSZ12] for the special case of uniform monadic predicates. We will rely on those characterizations to obtain several properties of the advice regular languages. Our main goal is the following regularity question:

Given an advice regular language L, when is L regular?

To answer this question, we introduce two notions:

- The *substitution property*, which states that if a formula φ together with the predicate $\mathbf{P}$ defines a regular language $L_{\varphi, \mathbf{P}}$, then there exists a *regular predicate* $\mathbf{Q}$ such that $L_{\varphi, \mathbf{Q}} = L_{\varphi, \mathbf{P}}$.
- The *syntactical predicate* of a language L, which is the "simplest" predicate $\mathbf{P}_L$ such that $L \in \mathbf{MSO}[\leq, \mathbf{P}_L]$.

Our second contribution is to show that the class of advice regular languages has the substitution property, and that an advice regular language L is regular if, and only if, $\mathbf{P}_L$ is regular. We apply these results to the case of *morphic predicates* [CT02], and obtain the following decidability result: given a language defined by an $\mathbf{MSO}$ formula with morphic predicates, one can decide whether it is regular.

1.3 Motivations from Circuit Complexity

Extending logics with predicates also appears in the context of circuit complexity. Indeed, a descriptive complexity theory initiated by Immermann [Imm87] relates logics and circuits. For instance, a language is recognized by a Boolean circuit of constant depth and unlimited fan-in if, and only if, it can be described by a first-order formula with any numerical predicates of any arity, *i.e.* $\mathbf{AC^0} = \mathbf{FO}[\mathrm{Arb}]$.

This correspondence led to the study of two properties, which characterize the regular languages (Straubing Conjecture) and the languages with a neutral letter (Crane Beach Conjecture) in several fragments and extensions of $\mathbf{FO}[\mathrm{Arb}]$. The Straubing Conjecture would, if true, give a deep understanding of many complexity classes inside $\mathbf{NC^1}$. For instance the Straubing Conjecture for first-order logic with counting quantifiers is equivalent to the separation of $\mathbf{ACC}$ and $\mathbf{NC^1}$. In the case of two-variable first-order logic, it implies tight bounds for the addition function. Many cases of this conjecture are still open and are often equivalent to proving circuit lower bounds. The Crane

Beach Conjecture was introduced as a model-theoretic approach to prove lower bounds, however this conjecture has been disproved [BIL⁺05]. On the positive sides, both conjectures hold in the special case of monadic predicates [BIL⁺05, Str94] for several fragments. Our third contribution is to give simple proofs of both the Straubing and the Crane Beach Conjectures for monadic predicates, relying on our previous characterizations and extending them to abstract fragments. Recently, Gehrke et al [GKP16] studied first-order logic with monadic predicates but restricted to one variable, and were able to obtain equations characterizing the regular languages in this class.

1.4 Outline

Section 2 gives characterizations of advice regular languages, in automata-theoretic, algebraic and machine-independent terms. In Section 3, we study the regularity question, and give two different answers: one through the substitution property, and the other through the existence of a syntactical predicate. The last section, Section 4, provides applications of our results: easy proofs that the Straubing and the Crane Beach Conjectures hold for monadic predicates and decidability of the regularity problem for morphic regular languages.

2 ADVICE REGULAR LANGUAGES

In this section, we introduce the class of advice regular languages and give several characterizations.

2.1 Predicates

A numerical predicate $\mathbf{P}$ of arity k is given by $\mathbf{P} = (\mathbf{P}_n)_{n \in \mathbb{N}}$, where $\mathbf{P}_n \subseteq \{0, \dots, n-1\}^k$. Since we mostly deal with monadic numerical predicates, we often drop the word "monadic numerical". In this definition the predicates are non-uniform: for each length the predicate is different. A predicate $\mathbf{P}$ if uniform if it is a relation over the natural numbers. More formally, if there exists $\mathbf{Q} \subseteq \mathbb{N}^k$ such that for every n, $\mathbf{P}_n = \mathbf{Q} \cap \{0, \dots, n-1\}^k$. In this case we identify $\mathbf{P}$ and $\mathbf{Q}$, and see uniform predicates as subsets of $\mathbb{N}^k$.

Example 2.1. The predicate **first** $= (\{0\})_{n \in \mathbb{N}}$, which is true only on the *first* position, is uniform; we denote it by $\{0\}$. Similarly, the predicate **last** $= (\{n-1\})_{n \in \mathbb{N}}$, which is true only for the *last* position, is not uniform.

In this paper, we will often treat predicates as words, identifying $\mathbf{P} = (\mathbf{P}_n)_{n \in \mathbb{N}}$ with $\mathbf{P}_n \subseteq \{0, 1\}^n$. In this case we can see $\mathbf{P}$ as a language over $\{0, 1\}$, which contains exactly one word for each length. This simple idea is used throughout the paper, where logical formulae and automata treat predicates as words, allowing us to perform syntactical operations on them.

We often define predicates $\mathbf{P} = (\mathbf{P}_n)_{n \in \mathbb{N}}$ with $\mathbf{P}_n \in A^n$ for some finite alphabet A. This is not formally a predicate, but this amounts to defining one predicate for each letter in A, and this abuse of notation will prove very convenient. Similarly, any infinite word $w \in A^\omega$ can be seen as a uniform predicate.

Example 2.2. The predicate **first** can be seen as the infinite word 10^ω, and the predicate **last** as the language of finite words described by the regular expression 0^*1.

2.2 Monadic Second-Order Logic

The formulae we consider are monadic second-order (**MSO**) formulae, obtained from the following grammar:

$$\varphi \;=\; \mathbf{a}(x) \mid x \leq y \mid P(x) \mid \varphi \wedge \varphi \mid \neg\varphi \mid \exists x,\ \varphi \mid \exists X,\ \varphi\ .$$

Here $x, y, z, \dots$ are first-order variables, which will be interpreted by positions in the word, and $X, Y, Z, \dots$ are monadic second-order variables, which will interpreted by sets of positions in

the word. We say that $\mathbf{a}$ is a letter symbol, $\leq$ the ordering symbol and $P, Q, \ldots$ are the numerical monadic predicate symbols, often refered to as predicate symbols. The notation

$$\varphi(P^1, \ldots, P^\ell, x^1, \ldots, x^p, X^1, \ldots, X^q)$$

means that in φ, the predicate symbols are among $P^1, \ldots, P^\ell$, the free first-order variables are among $x^1, \ldots, x^p$ and the free second-order variables are among $X^1, \ldots, X^q$. A formula without free variables is called a sentence. We use the notation $\overline{P}$ to abbreviate $P^1, \ldots, P^\ell$, and similarly for all objects (variables, predicate symbols, predicates).

We now define the semantics. The letter symbols and the ordering symbol are always interpreted in the same way, as expected. For the predicate symbols, the predicate symbol P is interpreted by a predicate $\mathbf{P}$. Note that P is a syntactic object, while $\mathbf{P}$ is a predicate used as the interpretation of P. Consider a formula $\varphi(\overline{P}, \overline{x}, \overline{X})$, a finite word u of length n, predicates $\overline{\mathbf{P}}$ interpreting the predicate symbols from $\overline{P}$, a valuation $\overline{\mathbf{x}}$ of the free first-order variables and a valuation $\overline{\mathbf{X}}$ of the free second-order variables. We define $u, \overline{\mathbf{P}}, \overline{\mathbf{x}}, \overline{\mathbf{X}} \models \varphi$ by induction as usual, with

$$u, \overline{\mathbf{P}}, \overline{\mathbf{x}}, \overline{\mathbf{X}} \models P(y) \quad \text{if} \quad \mathbf{y} \in \mathbf{P}_n \ .$$

A sentence $\varphi(\overline{P})$ and a tuple of predicates $\overline{\mathbf{P}}$ interpreting the predicate symbols from $\overline{P}$ define a language

$$L_{\varphi, \overline{\mathbf{P}}} = \{u \in A^* \mid u, \overline{\mathbf{P}} \models \varphi\} \ .$$

Such a language is called advice regular, and the class of advice regular languages is denoted by $\mathrm{MSO}[\leq, \mathrm{Arb}_1]$.

2.3   Automata with Advice

We introduce automata with advice. Unlike classical automata, they have access to two more pieces of information about the word being read: its length and the current position. Both the transitions and the final states can depend on those two pieces of information. For this reason, automata with advice are (much) more expressive than classical automata, and recognize undecidable languages. A non-deterministic automaton with advice is given by $\mathcal{A} = (Q, q_0, \delta, F)$ where Q is a finite set of states, $q_0 \in Q$ is the initial state, $\delta \subseteq \mathbb{N} \times \mathbb{N} \times Q \times A \times Q$ is the transition relation and $F \subseteq \mathbb{N} \times Q$ is the set of final states. In the deterministic case, δ is a function from $\mathbb{N} \times \mathbb{N} \times Q \times A$ into Q.

A run of $\mathcal{A}$ over a finite word $u = u_0 \cdots u_{n-1} \in A^*$ is a finite word $\rho = q_0 \cdots q_n \in Q^*$ such that for all $i \in \{0, \ldots, n-1\}$, we have $(i, n, q_i, u_i, q_{i+1}) \in \delta$. It is accepting if $(n, q_n) \in F$. One obtains a uniform model by removing one piece of information in the transition function: the length of the word. This automaton model is strictly weaker, and is (easily proved to be) equivalent to the one introduced in [KRSZ12], where the automata read at the same time the input word and a fixed word called the advice. However, our definition will be better suited for some technical aspects: for instance, the number of Myhill-Nerode equivalence classes exactly corresponds to the number of states in a minimal deterministic automaton.

Example 2.3. The language $\{a^n b^n c^n \mid n$ is a prime number$\}$ is recognized by a (deterministic) automaton with advice. The automaton is represented in Figure 1. It has five states, q_a, q_b, q_c, q_F
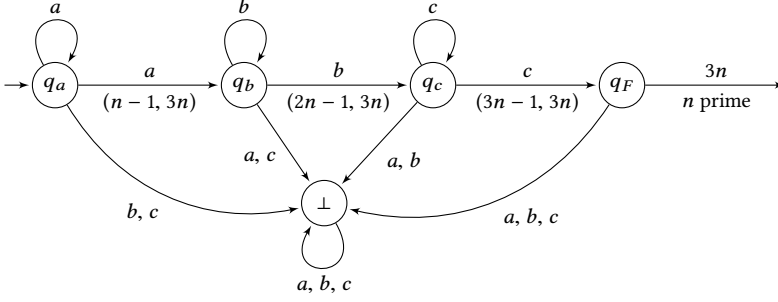
Fig. 1. The automaton for Example 2.3.

and $\perp$. The initial state is q_a. The transition function is defined as follows:

$$
\begin{array}{rcll}
\delta(i, 3n, q_a, a) & = & q_a & \text{if } i < n - 1 \\
\delta(n - 1, 3n, q_a, a) & = & q_b & \\
\delta(i, 3n, q_b, b) & = & q_b & \text{if } n \leq i < 2n - 1 \\
\delta(2n - 1, 3n, q_b, c) & = & q_c & \\
\delta(i, 3n, q_c, c) & = & q_c & \text{if } 2n \leq i < 3n - 1 \\
\delta(3n - 1, 3n, q_c, c) & = & q_F &
\end{array}
$$

All other transitions lead to $\perp$, the sink rejecting state. The set of final states is $F = \{(3n, q_F) \mid n \text{ is a prime number}\}$.

We mention another example, that appeared in the context of automatic structures. The paper [Nie07] shows that the structure $(\mathbb{Q}, +)$ is automatic with advice, which amounts to showing that the language

$$
\{\widehat{x} \,\sharp\, \widehat{y} \,\sharp\, \widehat{z} \mid z = x + y\} \ ,
$$

where $\widehat{x}$ denotes the factorial representation of the rational x, is advice regular. A very difficult proof shows that this is not possible without advice [Tsa11].

2.4 One-scan Programs

Programs over monoids were introduced in the context of circuit complexity [Bar89]: Barrington showed that any language in $\mathbf{NC}^1$ can be computed by a program of polynomial length over a non-solvable group. We present a simplification adapted to monadic predicates, introduced in [Str92] and developed in [BS95]. We refer to [Str94, Chapter IX.4] for a complete presentation. In these works, Barrington and Straubing use Ramsey-theoretic methods to obtain non-expressibility results. In the remainder of this paper, we will show a generalization of these results, using a syntactic approach. In particular we avoid the use of Ramsey type arguments.

A one-scan program is given by $P = (M, (f_{i,n} : A \to M)_{i,n \in \mathbb{N}}, S)$ where M is a finite monoid and $S \subseteq M$. The function $f_{i,n}$ is used to compute the effect of the ith letter of an input word of length n. We say that the program P accepts the word $u = u_0 \cdots u_{n-1}$ if

$$
f_{0,n}(u_0) \cdots f_{n-1,n}(u_{n-1}) \in S \ .
$$

Note that this echoes the classical definition of recognition by monoids, where a morphism $f : A \to M$ into a finite monoid M recognizes the word $u = u_0 \cdots u_{n-1}$ if $f(u_0) \cdots f(u_{n-1}) \in S$. Here, a one-scan program uses different functions $f_{i,n}$, depending on the position i and the length of the word n.

Example 2.4. Let U_1 be the monoid over $\{0, 1\}$ equipped with the classical multiplication. Consider the alphabet $A = \{a, b\}$. We define the one-scan program $(U_1, (f_{i,n})_{i,n \in \mathbb{N}}, \{1\})$ as follows. If n is not a prime number, then $f_{i,n}$ is constant equal to 0. Otherwise, $f_{i,n}(a) = 0$ and $f_{i,n}(b) = 1$. Therefore, a word is accepted by this one-scan program if, and only if, its length is prime and all prime positions are labelled by the letter b.

2.5 Myhill-Nerode Equivalence Relations

Let $L \subseteq A^*$ and $p \in \mathbb{N}$, we define two equivalence relations:

- $u \sim_L v$ if for all $w \in A^*$, we have $uw \in L \Longleftrightarrow vw \in L$,
- $u \sim_{L,p} v$ if for all $w \in A^p$, we have $uw \in L \Longleftrightarrow vw \in L$.

The relation $\sim_L$ is called the (classical) Myhill-Nerode equivalence relation, and the second is a coarser relation, which we call the p-Myhill-Nerode equivalence relation. Recall that $\sim_L$ contains finitely many equivalence classes if, and only if, L is regular, *i.e.* $L \in \mathbf{MSO}[\leq]$.

2.6 Equivalence

We state several characterizations of advice regular languages.

THEOREM 2.5 (ADVICE REGULAR LANGUAGES). *Let L be a language of finite words. The following properties are equivalent:*

(1) $L \in \mathbf{MSO}[\leq, \mathrm{Arb}_1]$,
(2) *L is recognized by a non-deterministic automaton with advice,*
(3) *L is recognized by a deterministic automaton with advice,*
(4) *There exists $K \in \mathbb{N}$ such that for all $i, p \in \mathbb{N}$, the restriction of $\sim_{L,p}$ to words of length i contains at most K equivalence classes.*
(5) *L is recognized by a one-scan program,*

In this case, we say that L is advice regular.

This extends the Myhill-Nerode theorem proposed in [KRSZ12], which proves the equivalence between (3) and (5) for the special case of uniform predicates.

PROOF. The implication $2 \Rightarrow 3$ is proved by determinizing automata with advice, extending the classical powerset construction. Let $\mathcal{A} = (Q, q_0, \delta, F)$ be a non-deterministic automaton with advice. We construct the deterministic automaton with advice $\mathcal{A}' = (Q', \{q_0\}, \delta', F')$, where Q' is the powerset of Q, the set of final states is $F' = \{(n, S) \mid \exists q \in S, (n, q) \in F\}$, and the transition function δ' is defined by:

$$\delta'(i, n, S, a) = \{q' \in Q \mid \exists q \in S, (i, n, q, a, q') \in \delta\}.$$

It is easy to see that $\mathcal{A}$ and $\mathcal{A}'$ are equivalent.

The implication $3 \Rightarrow 2$ is immediate from the definitions. The implication $1 \Rightarrow 2$ requires us to show closure properties of automata with advice under union, projection and complementation. The first two closures are obtained in the exact same way as in the classical case, we do not detail them here; for the third case, we rely on the equivalence between 3 and 4, and complement deterministic (complete) automata with advice by simply exchanging F and its complement in $\mathbb{N} \times Q$.

The implication $2 \Rightarrow 1$ amounts to writing a formula checking for the existence of a run. Let $\mathcal{A} = (Q, q_0, \delta, F)$ be a non-deterministic automaton with advice recognizing a language L.

$$\exists \overline{X}, \begin{cases} \chi(\overline{X}) \\ \wedge \, \forall x, \quad \text{first}(x) \implies x \in X_{q_0} \\ \wedge \, \forall x, \forall y, \quad y = x + 1 \wedge \bigwedge_{(q,a) \in Q \times A} \bigvee_{q' \in Q} \\ \qquad T^{q,a,q'}(x) \wedge x \in X_q \wedge a(x) \wedge y \in X_{q'} \\ \wedge \, \forall x, \quad \text{last}(x) \implies \bigvee_{q \in Q} \quad x \in X_q \wedge F^q(x) \, . \end{cases}$$

Fig. 2. Formula checking for the existence of an accepting run.

Let $\overline{X}$ be a Q-tuple of monadic second-order variables. We first need to express that $\overline{\mathbf{X}}$ partitions the set of all positions of the word. This is easily expressed by the following formula, denoted $\chi(\overline{X})$:

$$\forall x, \left(\bigvee_{q \in Q} x \in X_q \right) \wedge \bigvee_{q \in Q} \left(x \in X_q \to \bigwedge_{q' \neq q \in Q} x \notin X_{q'} \right) \, .$$

For each $q \in Q$, we define the predicates $\mathbf{T}^{q,a,q'}$ by $\mathbf{T}_n^{q,a,q'} = \{i \in \mathbb{N} \mid \delta(i, n, q, a) = q'\}$ and $\mathbf{F}^q$ by $\mathbf{F}^q = \{n \in \mathbb{N} \mid (n, q) \in F\}$.

The **MSO** formula φ in figure 2 checks for the existence of an accepting run, and uses the predicate symbols $T^{q,a,q'}$ and F^q. We have $L_{\varphi, \{T^{q,a,q'}, F^q\}} = L$.

For the implication $3 \Rightarrow 4$, let $\mathcal{A}$ be a deterministic automaton with advice. Let $n = i + p$, and consider the mapping $t_{i,n} : A^i \to Q$ defined for $u = u_0 \cdots u_{i-1}$ by

$$t_{i,n}(u) = \delta(i - 1, n, \delta(i - 2, n, \cdots \delta(0, n, q_0, u_0) \cdots , u_{i-2}), u_{i-1}) \, .$$

In words, $t_{i,n}(u)$ is the state reached by $\mathcal{A}$ while reading u of length i assuming that the total word will be of length n. We argue that $t_{i,n}(u) = t_{i,n}(v)$ implies $u \sim_{L,p} v$: indeed, for $w \in A^p$, after reading u or v, the automaton $\mathcal{A}$ is in the same state, so it will either accept both uw and vw or reject both. Note that $t_{i,n}$ can have at most $|Q|$ different values. Consequently, the restriction of $\sim_{L,p}$ to words of length i contains at most $|Q|$ equivalence classes.

We now prove the implication $4 \Rightarrow 3$, by constructing a deterministic automaton with advice. Its set of states is $Q = \{0, \ldots, K - 1\}$. To each word u and length $n \geq 0$ we associate $\lfloor u \rfloor_n \in Q$ such that if u and v both have length i, then $u \sim_{L,n-i} v$ if, and only if, $\lfloor u \rfloor_n = \lfloor v \rfloor_n$. We set $\lfloor \varepsilon \rfloor_n = 0$ for all n; the initial state is 0. The transition function is defined by $\delta(i, n, \lfloor u \rfloor_n, a) = \lfloor ua \rfloor_n$, for *some* u of length i. (This is well defined: if both u and v have length i and $\lfloor u \rfloor_n = \lfloor v \rfloor_n$, then $u \sim_{L,n-i} v$, so $ua \sim_{L,n-i-1} va$, thus $\lfloor ua \rfloor_n = \lfloor va \rfloor_n$.) The set of final states is

$$F = \{(n, \lfloor u \rfloor_0) \mid u \in A^n \cap L\} \, .$$

We argue that this automaton recognizes L: whenever it reads $u = u_0 \cdots u_{n-1}$, the corresponding run is

$$\rho = \lfloor \varepsilon \rfloor_n \lfloor u_0 \rfloor_n \lfloor u_0 u_1 \rfloor_n \cdots \lfloor u_0 \cdots u_{n-1} \rfloor_n \, ,$$

which is accepting if, and only if, $u \in L$.

The implication $3 \Rightarrow 5$ is syntactical. Consider a deterministic automaton with advice $\mathcal{A} = (Q, q_0, \delta, F)$ recognizing a language L. We define M to be the monoid of functions from Q to Q, with composition as multiplication. Define

$$f_{i,n} : \begin{cases} A \to M \\ a \mapsto (q \mapsto \delta(i, n, q, a)) \, , \end{cases}$$

and $S = \{\phi \in M \mid \phi(q_0) \in F\}$. The one-scan program $(M, (f_{i,n})_{i,n \in \mathbb{N}}, S)$ recognizes L.

The converse implication 5 ⇒ 3 is also syntactical.

Consider a one-scan program $(M, (f_{i,n})_{i,n \in \mathbb{N}}, S)$ recognizing a language L. Define the deterministic automaton with advice $\mathcal{A} = (M, 1, \delta, F)$ where 1 is the neutral element of M, the transition function δ is defined by $\delta(i, n, m, a) = m \cdot f_{i,n}(a)$, and $F = \{(n, m) \mid m \in S\}$. The automaton $\mathcal{A}$ recognizes the language L.   □

3 THE REGULARITY QUESTION

In this section, we address the following question: given an advice regular language, when is it regular? We answer this question in two different ways: first by showing a substitution property, and second by proving the existence of a syntactical predicate. Note that the regularity question is not a decision problem, as advice regular languages are not finitely presentable, so we can only provide non-effective characterizations of regular languages inside this class.

In the next section, we will show applications of these two notions: first by proving that the Straubing property holds in this case, and second by proving the decidability of the regularity problem for morphic regular languages.

3.1 A Substitution Property

In this subsection, we prove a substitution property for $\mathbf{MSO}[\leq, \mathrm{Arb}_1]$ and for $\mathbf{MSO}[\leq, \mathrm{Arb}_1^{\mathrm{u}}]$. We start by defining the class of (monadic) *regular predicates*.

The predicates $\{c\}$ and $\mathbf{last} - c = (\{n - 1 - c\})_{n \in \mathbb{N}}$ for a given $c \in \mathbb{N}$ are called local predicates. The predicates $\{x \mid x \equiv r \bmod q\}$ and $\mathbf{last} \equiv r \bmod q$ for given $q, r \in \mathbb{N}$ are called modular predicates.

THEOREM 3.1 ([PÉL92, STR94]). *Let* $\mathbf{P} = (\mathbf{P}_n)_{n \in \mathbb{N}}$ *be a predicate. The following properties are equivalent:*

(1) *There exists a formula* $\varphi(x) \in \mathbf{MSO}[\leq]$ *over the one-letter alphabet* $\{a\}$ *such that* $\mathbf{P}_n = \{\mathbf{x} \in \{0, n - 1\} \mid a^n, \mathbf{x} \models \varphi(x)\}$.
(2) *The predicate* $\mathbf{P}$ *is a boolean combination of local and modular predicates.*
(3) *The language* $\mathbf{P} \subseteq A^*$ *is regular.*

In this case, we say that $\mathbf{P}$ *is regular. We denote by* $\mathcal{R}eg_1$ *the class of regular (monadic) predicates.*

The following theorem states the substitution property for $\mathbf{MSO}[\leq, \mathrm{Arb}_1]$.

THEOREM 3.2. *For all sentences* $\varphi(\overline{P})$ *in* $\mathbf{MSO}[\leq, \mathrm{Arb}_1]$ *and predicates* $\overline{\mathbf{P}} \in \mathrm{Arb}_1$ *such that* $L_{\varphi, \overline{\mathbf{P}}}$ *is regular, there exist* $\overline{\mathbf{Q}} \in \mathcal{R}eg_1$ *such that* $L_{\varphi, \overline{\mathbf{Q}}} = L_{\varphi, \overline{\mathbf{P}}}$.

The main idea of the proof is that among all predicates $\overline{\mathbf{Q}}$ such that $L_{\varphi, \overline{\mathbf{P}}} = L_{\varphi, \overline{\mathbf{Q}}}$, there is a minimal one with respect to a lexicographic ordering, which can be defined by an $\mathbf{MSO}$ formula. The key technical point is given by the following lemma, which can be understood as a regular choice function.

LEMMA 3.3 (REGULAR CHOICE LEMMA). *Let* M *be a regular language such that for all* $n \in \mathbb{N}$, *there exists a word* $w \in M$ *of length* n. *Then there exists a regular language* $M' \subseteq M$ *such that for all* $n \in \mathbb{N}$, *there exists exactly one word* $w \in M'$ *of length* n.

PROOF. We equip the alphabet A with a total order, inducing the lexicographic ordering $\leq$ on A^*.

Let ψ be an $\mathbf{MSO}$ formula defining M. The objective is to define an $\mathbf{MSO}$ formula Ψ such that w satisfies Ψ if, and only if, w is minimal among the words of its length to satisfy ψ with respect to $\leq$. The language defined by this formula satisfies the desired properties.

First, let $\overline{X}$ be a A-tuple of monadic second-order variables. We say that $\overline{\mathbf{X}} \in A^n$ represents the word $v \in A^n$ if $\overline{\mathbf{X}}$ partitions the set of all positions and for all $i \in \{0, \dots, n - 1\}, a \in A$, we have

$v_i = a$ if, and only if, $i \in \mathbf{X}_a$. The formula expressing that $\overline{\mathbf{X}}$ partitions the set of all positions is denoted by $\chi(\overline{X})$ (see the proof of Theorem 2.5 for the definition of this formula).

We obtain a formula $\varphi(\overline{X})$ from ψ by syntactically replacing in ψ each letter predicate $\mathbf{a}(x)$ by $x \in X_a$. The following property holds: for all $\overline{\mathbf{X}} \in A^*$, if $\overline{\mathbf{X}}$ represents the word v, then $\overline{\mathbf{X}} \models \varphi(\overline{X})$ is equivalent to $v \models \psi$.

Now, we define a formula $\theta(\overline{X})$ such that for all words $w \in A^*$ and $\overline{\mathbf{X}} \in A^*$, if $\overline{\mathbf{X}}$ represents a word v, then $w, \overline{\mathbf{X}} \models \theta(\overline{X})$ if, and only if, $w \leq v$. There are two cases: either $w = v$, or $w \prec v$, so the formula $\theta(\overline{X})$ is a disjunction of two formulae, the first stating that $\overline{\mathbf{X}}$ represents w:

$$\forall x, \bigwedge_{a \in A} (x \in X_a \iff \mathbf{a}(x)) \ ,$$

and the second stating that $w \prec v$, where v is represented by $\overline{\mathbf{X}}$:

$$\exists x, \left(\bigvee_{a < b \in A} \mathbf{a}(x) \wedge x \in X_b \right) \wedge \left(\forall y, \ y < x \rightarrow \left(\bigvee_{a \in A} \mathbf{a}(y) \wedge y \in X_a \right) \right) .$$

The **MSO** formula Ψ that selects the minimal word in M is given by:

$$\psi \ \wedge \ \forall \overline{X}, \ \left(\chi(\overline{X}) \wedge \varphi(\overline{X}) \right) \rightarrow \theta(\overline{X}) .$$

This concludes the proof. □

We now prove Theorem 3.2 relying on Lemma 3.3.

PROOF. Consider $\varphi(\overline{P})$ a sentence and $\overline{\mathbf{P}}$ predicates such that $L_{\varphi, \overline{\mathbf{P}}}$ is regular. We write L for $L_{\varphi, \overline{\mathbf{P}}}$. Let θ be an **MSO** formula defining L.

Consider the language $M = \{ \overline{\mathbf{X}} \in (\{0,1\}^\ell)^* \mid L_{\varphi, \overline{\mathbf{X}}} = L \}$.

We first argue that M is regular.

As in the proof of Lemma 3.3, we introduce $\overline{Y}$ a A-tuple of monadic second-order variables, used to represent words in A^*. We obtain a formula $\psi(\overline{Y})$ from θ by syntactically replacing in θ each letter predicate $\mathbf{a}(x)$ by $x \in X_a$. The following property holds: for all $\overline{\mathbf{Y}} \in A^*$, if $\overline{\mathbf{Y}}$ represents the word v, then $\overline{\mathbf{Y}} \models \psi(\overline{Y})$ is equivalent to $v \models \theta$.

Consider the following formula in **MSO**[$\leq$] over the alphabet $\{0,1\}^\ell$:

$$\forall \overline{Y}, \ \chi(\overline{Y}) \implies \left(\varphi(\overline{X}) \iff \psi(\overline{Y}) \right) ,$$

it describes the language M: the word $\overline{\mathbf{X}} \in (\{0,1\}^\ell)^n$ satisfies this formula if for any word $v \in A^n$ represented by $\overline{\mathbf{Y}}$, v is in L if, and only if, $v, \overline{\mathbf{X}} \models \varphi(\overline{X})$.

Now, we note that for all $n \in \mathbb{N}$, there exists a word in M of length n, namely $\overline{\mathbf{P}}_n$. Thus Lemma 3.3 applies, so there exists $M' \subseteq M$ a regular language so that for all $n \in \mathbb{N}$, there exists a unique word in M' of length n, which we denote by $\overline{\mathbf{Q}}_n$. Thanks to Theorem 3.1, this yields a tuple of regular predicates $\overline{\mathbf{Q}}$ such that $L_{\varphi, \overline{\mathbf{Q}}} = L_{\varphi, \overline{\mathbf{P}}}$. □

We proved the substitution property for **MSO**[$\leq$, Arb$_1$]. We now prove that it also holds for **MSO**[$\leq$, Arb$_1^\mathrm{u}$]; note that this is not implied by the previous case. We first prove it over infinite words, and then transfer the result to finite words.

THEOREM 3.4. *For all sentences $\varphi(\overline{P})$ in* **MSO**[$\leq$, Arb$_1^\mathrm{u}$] *and predicates $\overline{\mathbf{P}} \in$ Arb$_1^\mathrm{u}$ such that $L_{\varphi, \overline{\mathbf{P}}}$ is ω-regular, there exist regular predicates $\overline{\mathbf{Q}} \in$ Arb$_1^\mathrm{u}$ such that $L_{\varphi, \overline{\mathbf{P}}} = L_{\varphi, \overline{\mathbf{Q}}}$.*

PROOF. Consider $\varphi(\overline{P})$ a sentence in $\mathbf{MSO}[\leq, \mathrm{Arb}_1^{\mathrm{u}}]$ and $\overline{P}$ predicates such that $L_{\varphi, \overline{P}}$ is ω-regular, denote it L.

Consider the following language:

$$M = \{\overline{\mathbf{X}} \in (\{0, 1\}^\ell)^\omega \mid L_{\varphi, \overline{X}} = L\} .$$

Relying on the same arguments as in the proof of Theorem 3.2, we show that M is ω-regular. It is also non-empty since it contains $\overline{\mathbf{P}}$. It follows from Büchi's Theorem that it contains a ultimately periodic word $\overline{\mathbf{Q}}$. Seen as predicates, $\overline{\mathbf{Q}}$ are regular monadic uniform predicates, and $L_{\varphi, \overline{P}} = L_{\varphi, \overline{Q}}$, which concludes the proof.   □

The following theorem states the substitution property for $\mathbf{MSO}[\leq, \mathrm{Arb}_1^{\mathrm{u}}]$.

THEOREM 3.5. *For all sentences $\varphi(\overline{P})$ in $\mathbf{MSO}[\leq, \mathrm{Arb}_1^{\mathrm{u}}]$ and predicates $\overline{P} \in \mathrm{Arb}_1^{\mathrm{u}}$ such that $L_{\varphi, \overline{P}}$ is regular, there exist regular predicates $\overline{Q} \in \mathrm{Arb}_1^{\mathrm{u}}$ such that $L_{\varphi, \overline{P}} = L_{\varphi, \overline{Q}}$.*

PROOF. We consider $\varphi(\overline{P})$ a sentence in $\mathbf{MSO}[\leq, \mathrm{Arb}_1^{\mathrm{u}}]$ and predicates $\overline{P}$ such that $L_{\varphi, \overline{P}}$ is regular. Let $\flat$ be a fresh letter (not in A), we denote by $A_\flat$ the alphabet $A \cup \{\flat\}$. We explain how to transform $\varphi(\overline{P})$ into a formula $\widehat{\varphi}(\overline{P})$ in $\mathbf{MSO}[\leq, \mathrm{Arb}_1^{\mathrm{u}}]$ over the alphabet $A_\flat$ satisfying: for all u in A^* and predicates $\overline{Q}$, we have:

$$u, \overline{Q} \models \varphi \iff u \cdot \flat^\omega, \overline{Q} \models \widehat{\varphi} \tag{1}$$

$\widehat{\varphi}(\overline{P})$ is obtained from $\varphi(\overline{P})$ by guarding every first-order quantifiers: the subformula $\exists y, \theta(y)$ is turned into the subformula $\exists y, \neg\flat(y) \wedge \widehat{\theta}(y)$. The equivalence (1) is easily proved by induction. Consider the formula $\psi(\overline{P})$ defined by

$$\widehat{\varphi}(\overline{P}) \ \wedge \ \exists y, \quad (\forall x \geq y, \ \flat(x) \ \wedge \ \forall x < y, \ \neg\flat(x))$$

and the predicates $\overline{P}$, they define the language $L_{\varphi, \overline{P}} \cdot \flat^\omega$ thanks to the equivalence (1), so it is ω-regular.

From Theorem 3.4, we get a tuple of regular predicates $\overline{Q} \in \mathrm{Arb}_1^{\mathrm{u}}$ such that $L_{\psi, \overline{Q}} = L_{\varphi, \overline{P}} \cdot \flat^\omega$. It follows that $L_{\varphi, \overline{P}} = L_{\varphi, \overline{Q}}$, which concludes the proof.   □

We note that the substitution property does not hold over binary predicates. In fact, one can show much worse: given M a deterministic Turing machine, one can construct a universal formula $\varphi_M(\overline{P})$ (*i.e.* with only universal quantifiers) with binary predicates such that $L_{\varphi_M, \overline{P}} = a^*$ if and only if $\overline{P}$ represents the run of M. In other words, even if the language of the formula is rather simple, it can use its predicates to perform arbitrarily complicated computations.

3.2 The Syntactical Predicate

In this subsection, we define the notion of syntactical predicate for an advice regular language. The word "syntactical" here should be understood in the following sense: the syntactical predicate $\mathbf{P}_L$ of L is the most regular predicate that describes the language L. In particular, we will prove that L is regular if, and only if, $\mathbf{P}_L$ is regular.

Let L be an advice regular language. We define the predicate $\mathbf{P}_L = (\mathbf{P}_{L, n})_{n \in \mathbb{N}}$. Thanks to Theorem 2.5, there exists $K \in \mathbb{N}$ such that for all $i, p \in \mathbb{N}$, the restriction of $\sim_{L, p}$ to words of length i contains at most K equivalence classes. Denote $Q = \{0, \ldots, K - 1\}$ and $\Sigma = (Q \times A \to Q) \uplus Q$, where $Q \times A \to Q$ is the set of functions from $Q \times A$ to Q. We define $\mathbf{P}_{L, n} \in \Sigma^n$.

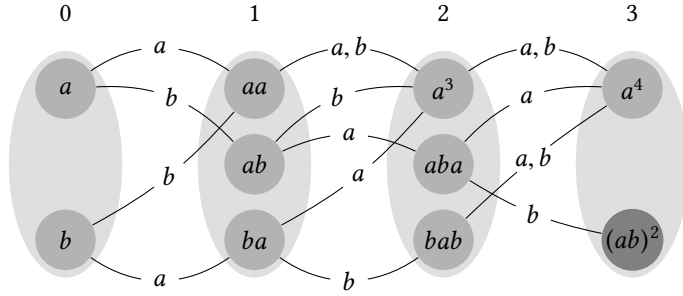Fig. 3. The predicate $\mathbf{P}_L$ (here $\mathbf{P}_{L,4}$) for $L = (ab)^* + (ba)^*b$.

Let $i, n \in \mathbb{N}$. Among all words of length i, we denote by $u_1^{i,n}, u_2^{i,n}, \ldots$ the lexicographically minimal representatives of the equivalence classes of $\sim_{L,n-i}$, enumerated in the lexicographic order:

$$u_1^{i,n} <_{\text{lex}} u_2^{i,n} <_{\text{lex}} u_3^{i,n} <_{\text{lex}} \cdots \tag{2}$$

In other words, $u_q^{i,n}$ is minimal with respect to the lexicographic order $<_{\text{lex}}$ among all words of length i in its equivalence class for $\sim_{L,n-i}$. Thanks to Theorem 2.5, there are at most K such words for each $i, n \in \mathbb{N}$.

We define $\mathbf{P}_{L,n}(i)$ (the ith letter of $\mathbf{P}_{L,n}$) by:

$$\mathbf{P}_{L,n}(i)(q, a) = q' \quad \text{if} \quad u_q^{i,n} \cdot a \sim_{L,n-i-1} u_{q'}^{i+1,n} \, , \text{ for } i < n-1 \tag{3}$$

$$\mathbf{P}_{L,n}(n-1)(q) \quad \text{if} \quad u_q^{n,n} \in L \, . \tag{4}$$

Intuitively, the predicate $\mathbf{P}_L$ describes the transition function with respect to the equivalence relations $\sim_{L,p}$. We now give an example.

Example 3.6. Consider the language $L = (ab)^* + (ba)^*b$. We represent $\mathbf{P}_{L,4}$ in figure 3. Each circle represents an equivalence class with respect to $\sim_{L,4}$, inside words of a given length. For instance, there are three equivalence classes for words of length 3: a^3, aba and bab. Note that these three words are the minimal representatives of their equivalence classes with respect to the lexicographic order. For the last position (here 3), the equivalence class of $(ab)^2$ (which is actually reduced to $(ab)^2$ itself) is darker since it belongs to the language L.

We state the main property of the predicate $\mathbf{P}_L$.

THEOREM 3.7. *Let L be an advice regular language. Then L is regular if, and only if, $\mathbf{P}_L$ is regular.*

The proof is split in two lemmas, giving each direction. We start by the *if* direction.

LEMMA 3.8. *Let L be an advice regular language. Then $L \in \mathbf{MSO}[\leq, \mathbf{P}_L]$.*

PROOF. From the definition of $\mathbf{P}_L$, it is easy to see that the word u of length n belongs to L if, and only if, there exists $X : \{0, \ldots, n\} \to Q$ such that:

$$\begin{aligned}
& \forall q \in Q, & X(0) = q &\iff \mathbf{P}_{L,n}(0)(0, u_0) = q \\
\wedge \quad & \forall q, q' \in Q, \forall i < n-1, & X(i+1) = q' &\impliedby X(i) = q \wedge \mathbf{P}_{L,n}(i)(q, u_i) = q' \\
\wedge \quad & \forall q \in Q, & X(n) = q &\implies \mathbf{P}_{L,n}(n-1)(q) \, .
\end{aligned}$$

This can be written down as an **MSO** formula with the predicate $\mathbf{P}_L$. □

The *if* direction of Theorem 3.7 follows from Lemma 3.8, because if $\mathbf{P}_L$ is regular, then $L \in \mathbf{MSO}[\leq, \mathbf{P}_L] = \mathbf{MSO}[\leq]$, so L is regular.

For the *only if* direction, we prove a stronger statement that will be useful in Section 4.4. Informally, we prove that the syntactic predicate $\mathbf{P}_L$ is the *least* predicate required to define in **MSO** the language L.

LEMMA 3.9. *Let L be an advice regular language defined with the predicates $\overline{\mathbf{P}}$. Then $\mathbf{P}_L \in \mathbf{MSO}[\leq, \overline{\mathbf{P}}]$.*

PROOF. Assume that L is defined by an **MSO** formula θ with the predicates $\overline{\mathbf{P}}$. Then the three equations (2), (3) and (4) defining $\mathbf{P}_L$ can be written down as an **MSO** formula with the predicates $\overline{\mathbf{P}}$.

To this end, we represent words as monadic second-order variables as in the proof of Lemma 3.3. A A-tuple $\overline{X}$ of monadic second-order variables represents the word $v \in A^n$ if $\overline{\mathbf{X}}$ partitions the set of all positions up to position n, and for all $i \in \{0, \ldots, n-1\}$, we have $v_i = a$ if, and only if, $i \in \mathbf{X}_a$.

Denote by $\chi(\overline{X}, x)$ the **MSO** formula expressing that $\overline{\mathbf{X}}$ partitions the set of all positions up to position $\mathbf{x}$. Similarly, denote by $\Xi(\overline{X}, x)$ the **MSO** formula expressing that $\overline{\mathbf{X}}$ partitions the set of all positions from the position $\mathbf{x} + 1$.

The formulae for (2) and (4) make use of the formulae $\chi(\overline{X}, x)$ and θ. We omit them as they are easy to write down, and focus on (3).

The first step is to construct a formula $\varphi(\overline{X}, \overline{Y}, w)$ such that if w has length n, $\overline{\mathbf{X}}$ represents u and $\overline{\mathbf{Y}}$ represents v both of length i, then $w, \overline{\mathbf{X}}, \overline{\mathbf{Y}}, i \models \varphi(\overline{X}, \overline{Y}, x)$ if, and only if, $u \sim_{L, n-i} v$. Define $\varphi(\overline{X}, \overline{Y}, x)$ as:

$$\forall \overline{Z}, \ \Xi(\overline{Z}, x) \implies \left(\phi(\overline{X}, x, \overline{Z}) \iff \phi(\overline{Y}, x, \overline{Z}) \right),$$

where the formula $\phi(\overline{X}, x, \overline{Z})$ is obtained from θ by syntactically replacing in θ each letter predicate $\mathbf{a}(y)$ by $(y \leq x \wedge y \in X_a) \vee (y > x \wedge y \in Z_a)$.

The second step is to construct a finite number of formulae $\gamma_\ell(\overline{X}, x)$ for $q \in Q$ such that if w has length n and $\overline{\mathbf{X}}$ represents u of length i, then $w, \overline{\mathbf{X}}, i \models \gamma_\ell(\overline{X}, x)$ if, and only if, there are exactly $q - 1$ words of length i that are (i) pairwise not equivalent with respect to $\sim_{L, n-i}$, (ii) not equivalent to u with respect to $\sim_{L, n-i}$, and (iii) smaller than u with respect to the lexicographic order.

We can now put the pieces together and give a formula for (3):

$$\forall x, \forall \overline{X}, \forall \overline{Y}, \bigwedge_{a \in A, \ q, q' \in Q} x \in X_a \wedge \chi(\overline{X}, x) \wedge \chi(\overline{Y}, x) \implies$$

$$P_{L, q, a, q'}(z) \iff (\gamma_q(\overline{X}, x) \wedge \gamma_{q'}(\overline{Y}, x)).$$

It follows that $\mathbf{P}_L$ is definable in $\mathbf{MSO}[\leq, \overline{\mathbf{P}}]$.   □

4 APPLICATIONS

In this section we show several consequences of Theorem 2.5 (characterization of the advice regular languages), Theorem 3.2 (a substitution property for advice regular languages) and Theorem 3.7 (a syntactical predicate for advice regular predicates).

The first two applications are about two conjectures, the Straubing and the Crane Beach Conjectures, introduced in the context of circuit complexity. We first explain the motivations for these two conjectures, and show very simple proofs of both of them in the special case of monadic predicates.

The third application shows that one can determine, given an **MSO** formula with morphic predicates, whether it defines a regular language.

4.1 A Descriptive Complexity for Circuit Complexity Classes

We first quickly define some circuit complexity classes. The most important here is $\mathbf{AC^0}$, the class of languages defined by boolean circuits of bounded depth and polynomial size, and its subclass $\mathbf{LAC^0}$ where the circuits have linear size. From $\mathbf{AC^0}$, adding the modular gates gives rise to $\mathbf{ACC}$. Finally, the class of languages defined by boolean circuits of logarithmic depth, polynomial size and fan-in 2 is denoted by $\mathbf{NC^1}$. Separating $\mathbf{ACC}$ from $\mathbf{NC^1}$ remains a long-standing open problem.

One approach to better understand these classes is through descriptive complexity theory, giving a perfect correspondence between circuit complexity classes and logical formalisms. Unlike what we did so far, the logical formalisms involved in this descriptive complexity theory use predicates of any arity (we focused on predicates of arity one). A k-ary predicate $\mathbf{P}$ is given by $(\mathbf{P}_n)_{n\in\mathbb{N}}$, where $\mathbf{P}_n \subseteq \{0, \ldots, n-1\}^k$. We denote by Arb the class of all predicates, and by $\mathcal{R}eg$ the class of regular predicates as defined in [Str94].

We recall the notations for some of the classical classes of formulae: $\mathbf{FO}$ (first-order quantifiers), $\mathbf{FO + MOD}$ (first-order and modular quantifiers: $\exists^{r,q}x, \varphi(x)$ reads "the number of x satisfying $\varphi(x)$ is equal to r mod q"), $\mathbf{FO^2}$ (first-order with at most two variables) and $\mathcal{B}\Sigma_k$ (at most $k-1$ alternations of $\exists$ and $\forall$ quantifiers).

THEOREM 4.1 ([IMM87, BCST92, GL84, KLPT06]).

(1) $\mathbf{AC^0} = \mathbf{FO}[\text{Arb}]$,
(2) $\mathbf{LAC^0} = \mathbf{FO^2}[\text{Arb}]$,
(3) $\mathbf{ACC} = (\mathbf{FO + MOD})[\text{Arb}]$.

Two conjectures have been formulated on the logical side, which aim at clarifying the relations between different circuit complexity classes. They have been stated and studied in special cases, we extrapolate them here to all fragments. We first need to give an abstract notion of (logical) fragment. Several such notions can be found in the bibliography, with more or less strong syntactic restrictions (see [KL12]). In this paper, we use a minimalist definition of fragment: we only require to be allowed to substitute predicates within each formula. Remark that this property is not restrictive and is satisfied by all classical fragments of $\mathbf{MSO}$. We fix the universal signature, containing infinitely many predicate symbols for each arity. Let $\mathbf{F}$ be a class of formulae over this signature and $\mathcal{P}$ a class of predicates, describing the fragment $\mathbf{F}[\mathcal{P}]$ by:

$$\mathbf{F}[\mathcal{P}] = \{L_{\varphi,\overline{\mathbf{P}}} \mid \varphi \in \mathbf{F} \wedge \overline{\mathbf{P}} \in \mathcal{P}\} .$$

The first property, called the Straubing property, characterizes the regular languages (denoted by $\mathbf{REG}$) inside a larger fragment.

Definition 4.2 (Straubing Property). $\mathbf{F}[\mathcal{P}]$ has the Straubing property if: all regular languages definable in $\mathbf{F}[\mathcal{P}]$ are also definable in $\mathbf{F}[\mathcal{P} \cap \mathcal{R}eg]$.

In symbols,

$$\mathbf{F}[\mathcal{P}] \cap \mathbf{REG} = \mathbf{F}[\mathcal{P} \cap \mathcal{R}eg] .$$

This statement appears for the first time in [BCST92], where it is proved that $\mathbf{FO}[\text{Arb}]$ has the Straubing property, relying on lower bounds for $\mathbf{AC^0}$ and an algebraic characterization of $\mathbf{FO}[\mathcal{R}eg]$. Following this result, Straubing conjectures in [Str94] that $(\mathbf{FO + MOD})[\text{Arb}]$ and $\mathcal{B}\Sigma_k[\text{Arb}]$ have the Straubing property for $k \geq 1$. Recently, this conjecture has been extended to $\mathbf{FO^2}[\text{Arb}]$ (see [KLPT06]). If true, it would imply the separation of $\mathbf{ACC}$ from $\mathbf{NC^1}$, and for the $\mathbf{FO^2}$ case, tight lower bounds on the addition of two integers in binary.

We already mentioned that several several fragments have the Straubing property, as for instance, $\Sigma_1[\text{Arb}]$, $\mathbf{FO}[\leq, \text{Arb}_1]$ and $(\mathbf{FO + MOD})[\leq, \text{Arb}_1]$, as proved by Straubing and Barrington [Str92,

BS95] by using Ramsey arguments for one scan programs and algebraic characterizations of these fragments. In this paper, we give a simpler syntactical proof that all fragments $\mathbf{F}[\leq, \mathrm{Arb}_1]$ have the Straubing property. The second property, called the Crane Beach property, characterizes the languages having a neutral letter, and is derived from a conjecture proposed by Thérien for the special case of first-order logic and finally disproved in the article [BIL$^+$05].

Definition 4.3 (Neutral letter). A language L has a neutral letter $e \in A$ if for all words u, v, we have $uv \in L$ if, and only if, $uev \in L$.

Definition 4.4 (Crane Beach Property). $\mathbf{F}[\mathcal{P}]$ has the Crane Beach property if: all languages having a neutral letter definable in $\mathbf{F}[\mathcal{P}]$ are definable in $\mathbf{F}[\leq]$.

Unfortunately, as mentioned, the Crane Beach property does not hold in general.

THEOREM 4.5 ([BIL$^+$05, SCH01]). *There exists a non-regular language having a neutral letter definable in* $\mathbf{FO}[\mathrm{Arb}]$.

A deeper understanding of the Crane Beach property specialized to first-order logic can be found in [BIL$^+$05]. In particular, it has been shown that $\mathbf{FO}[\leq, \mathrm{Arb}_1]$ has the Crane Beach property. In this paper, we give a simple proof that $\mathbf{MSO}[\leq, \mathrm{Arb}_1]$ has the Crane Beach Property.

4.2 The Straubing Conjecture for Advice Regular Languages

THEOREM 4.6. *All fragments* $\mathbf{F}[\leq, \mathrm{Arb}_1]$ *have the Straubing property.*

This is actually a straightforward corollary of Theorem 3.2.

PROOF. Let $\varphi \in \mathbf{F}$ such that $L_{\varphi, \overline{\mathbf{P}}}$ with $\overline{\mathbf{P}} \in \mathrm{Arb}_1$ is regular. Thanks to Theorem 3.2, there exist $\overline{\mathbf{Q}} \in \mathcal{R}eg_1$ such that $L_{\varphi, \overline{\mathbf{Q}}} = L_{\varphi, \overline{\mathbf{P}}}$. This concludes the proof. □

We state a corollary of Theorem 4.6.

COROLLARY 4.7. *For all* $k \geq 1$, $\mathcal{B}\Sigma_k[\leq, \mathrm{Arb}_1]$ *has the Straubing property.*

We conclude this subsection by remarking that the substitution property does not hold over infinite words, even for monadic predicates. This follows from the simple observation that adding the "bit-predicate" to first-order logic allows us to express all of monadic second-order logic. Formally, the bit-predicate $\mathbf{B}$ is defined by $\mathbf{B}(x, y)$ holds if the yth bit of the binary representation of x is 1. Roughly speaking, in the setting of infinite words the bit-predicate can make use of the infinite number of positions to talk about any finite set, hence first-order logic with the bit-predicate expresses all of weak monadic second-order logic, which coincides with monadic second-order logic.

Now the Straubing Property over infinite words for first-order logic reads:

$$\mathbf{FO}[\mathrm{Arb}] \cap \omega\mathbf{REG} = \mathbf{FO}[\mathcal{R}eg] .$$

This would imply $\mathbf{MSO}[\leq] \subseteq \mathbf{FO}[\mathcal{R}eg]$, which does not hold: the parity language, defined by $L = \{u \cdot \natural^\omega \mid u \in \{a, b\}^*$ has an even number of $a\}$ belongs to $\mathbf{MSO}[\leq]$, but not to $\mathbf{FO}[\mathcal{R}eg]$ [STT95].

4.3 The Crane Beach Conjecture for Advice Regular Languages

In this subsection, we show that the Crane Beach Conjecture holds for advice regular languages.

THEOREM 4.8. $\mathbf{MSO}[\leq, \mathrm{Arb}_1]$ *has the Crane Beach property.*

The proof is a simple corollary of Theorem 2.5.

Proof. Recall that a language over finite words L has a neutral letter $e \in A$ if for all words u and v, we have $uv \in L$ if, and only if, $uev \in L$. In other words, $u \sim_L ue$.

Let L be an advice regular language, thanks to Theorem 2.5, there exists $K \in \mathbb{N}$ such that for all $i, p \in \mathbb{N}$, the restriction of $\sim_{L,p}$ to words of length i contains at most K equivalence classes.

We argue that $\sim_L$ contains at most K equivalence classes (without both restrictions to words of a given length). Indeed, assume to the contrary that there are $K + 1$ words that are pairwise non-equivalent with respect to $\sim_L$. By iterating the equivalence $u \sim_L ue$, we obtain $K + 1$ words of the same length (the maximal length of the $K + 1$ original words), which are still pairwise non-equivalent with respect to $\sim_L$. For two non-equivalent words u, v, there exist a third word w witnessing the non-equivalence: $uw \in L$ but $vw \notin L$ or the other way around. Again by padding with the neutral letter e, we obtain non-equivalence witnesses for each pair of the $K + 1$ words of the same length (the maximal length of the $\frac{(K+1)(K+2)}{2}$ witnesses). Hence we have $K + 1$ words of the same length which are not equivalent with respect to $\sim_{L,p}$ for the same p, a contradiction. It follows that L is regular, i.e $L \in \mathbf{MSO}[\leq]$. □

4.4 Morphic Regular Languages

In this subsection, we apply Theorem 3.7 to the case of morphic predicates, and obtain the following result: given an **MSO** formula with morphic predicates, it is decidable whether it defines a regular language.

The class of morphic predicates was first introduced by Thue in the context of combinatorics on words, giving rise to the HD0L systems. Formally, let A, B be two finite alphabets, $\sigma : A^* \to A^*$ a morphism, $a \in A$ a letter such that $\sigma(a) = a \cdot u$ for some $u \in A^+$ and $\varphi : A^* \to B^*$ a morphism. This defines the sequence of words $\varphi(a), \varphi(\sigma(a)), \varphi(\sigma^2(a)), \ldots$, which converges to a finite or infinite word. An infinite word obtained in this way is said to be morphic.

We see morphic words as predicates, and denote by HD0L the class of morphic predicates. We call the languages definable in $\mathbf{MSO}[\leq, \mathrm{HD0L}]$ morphic regular.

THEOREM 4.9. *The following problem is decidable: given L a morphic regular language, is L regular? Furthermore, if L is regular, then we can construct a finite automaton for L.*

The proof of this theorem goes in two steps:
- first, we reduce the regularity problem for a morphic regular language L to deciding the ultimate periodicity of $\mathbf{P}_L$,
- second, we show that $\mathbf{P}_L$ is morphic.

Hence we rely on the following result: given a morphic word, it is decidable whether it is ultimately periodic. The decidability of this problem was conjectured 30 years ago and proved recently and simultaneously by Durand and Mitrofanov [Dur13, Iva12].

The first step is a direct application of Theorem 3.7. For the second step, observe that thanks to Lemma 3.9, we have $\mathbf{P}_L \in \mathbf{MSO}[\leq, \mathrm{HD0L}]$. We conclude with the following lemma, which follows from the characterization of morphic words as being those automatically presentable with the lexicographic ordering [RM02].

LEMMA 4.10. HD0L *is closed under* **MSO**-*interpretations, i.e. if $\mathbf{P}$ is an infinite word such that $\mathbf{P} \in \mathbf{MSO}[\leq, \mathrm{HD0L}]$, then $\mathbf{P} \in \mathrm{HD0L}$.*

Furthermore, all constructions in this proof are effective, and if $\mathbf{P}_L$ is ultimately periodic, then one can compute the threshold and the period, and derive from them a finite automaton for L.

As a corollary, we also obtain from Theorem 4.9 the decidability of $\mathbf{MSO}[\leq, \mathrm{HD0L}]$. Indeed, from a language in $\mathbf{MSO}[\leq, \mathrm{HD0L}]$, we first determine whether it is regular, and: if it is regular,

then determine whether it is empty by looking at the (effectively constructed) finite automaton recognizing it, and if it is not regular, then it is non-empty (since the empty language is regular). We stress however that this result can be obtained with a much more direct proof [CT02].

ACKNOWLEDGMENTS

REFERENCES

[Bar89]   David A. Mix Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *Journal of Computer and System Sciences*, 38(1):150–164, 1989.

[BCST92] David A. Mix Barrington, Kevin Compton, Howard Straubing, and Denis Thérien. Regular languages in NC^1. *Journal of Computer and System Sciences*, 44(3):478–499, 1992.

[BIL+05] David A. Mix Barrington, Neil Immerman, Clemens Lautemann, Nicole Schweikardt, and Denis Thérien. First-order expressibility of languages with neutral letters or: The Crane Beach conjecture. *Journal of Computer and System Sciences*, 70(2):101–127, 2005.

[BS95]   David A. Mix Barrington and H. Straubing. Superlinear lower bounds for bounded-width branching programs. *Journal of Computer and System Sciences*, 50(3):374 – 381, 1995.

[Büc62]   Julius R. Büchi. On a decision method in restricted second-order arithmetic. In *CLMPS*, pages 1–11. Stanford University Press, 1962.

[CT02]   Olivier Carton and Wolfgang Thomas. The monadic theory of morphic infinite words and generalizations. *Information and Computation*, 176(1):51–65, 2002.

[Dur13]   Fabien Durand. Decidability of the HD0L ultimate periodicity problem. *RAIRO Theor. Inform. Appl.*, 47(2):201–214, 2013.

[ER66]   Calvin C. Elgot and Michael O. Rabin. Decidability and undecidability of extensions of second (first) order theory of (generalized) successor. *Journal of Symbolic Logic*, 31(2):169–181, 1966.

[FP14]   Nathanaël Fijalkow and Charles Paperman. Monadic second-order logic with arbitrary monadic predicates. In *MFCS*, pages 279–290, 2014.

[GKP16]   Mai Gehrke, Andreas Krebs, and Jean-Éric Pin. Ultrafilters on words for a fragment of logic. *Theoretical Computer Science*, 610:37–58, 2016.

[GL84]   Yuri Gurevich and Harry R. Lewis. A logic for constant-depth circuits. *Information and Control*, 61(1):65–74, 1984.

[Imm87]   Neil Immerman. Languages that capture complexity classes. *SIAM Journal of Computing*, 16(4):760–778, 1987.

[Iva12]   Mitrofanov Ivan. A proof for the decidability of HD0L ultimate periodicity. *arXiv:1110.4780*, 2012.

[KL12]   Manfred Kufleitner and Alexander Lauser. Lattices of logical fragments over words. In *ICALP*, pages 275–286, 2012.

[KLPT06] Michal Koucký, Clemens Lautemann, Sebastian Poloczek, and Denis Thérien. Circuit Lower Bounds via Ehrenfeucht-Fraissé Games. In *IEEE Conference on Computational Complexity*, pages 190–201, 2006.

[KRSZ12] Alex Kruckman, Sasha Rubin, John Sheridan, and Ben Zax. A Myhill-Nerode theorem for automata with advice. In *GandALF*, pages 238–246, 2012.

[Nie07]   André Nies. Describing groups. *Bulletin of Symbolic Logic*, 13:305–339, 9 2007.

[Pél92]   Pierre Péladeau. Logically defined subsets of $\mathbb{N}^k$. *Theoretical Computer Science*, 93(2):169–183, 1992.

[Rab07]   Alexander Rabinovich. On decidability of monadic logic of order over the naturals extended by monadic predicates. *Information and Computation*, 205(6):870–889, 2007.

[Rab12]   Alexander Rabinovich. The Church problem for expansions of $(\mathbb{N}, <)$ by unary predicates. *Information and Computation*, 218:1–16, 2012.

[RM02]   Michel Rigo and Arnaud Maes. More on generalized automatic sequences. *Journal of Automata, Languages and Combinatorics*, 7(3):351–376, 2002.

[RT06]   Alexander Rabinovich and Wolfgang Thomas. Decidable theories of the ordering of natural numbers with unary predicates. In *CSL*, pages 562–574, 2006.

[Sch01]   Nicole Schweikardt. *On the Expressive Power of First-Order Logic with Built-In Predicates*. PhD thesis, Gutenberg-Universität in Mainz, 2001.

[Sem84]   Alexei L. Semenov. Decidability of monadic theories. In *MFCS*, pages 162–175, 1984.

[Str92]   Howard Straubing. Circuit complexity and the expressive power of generalized first-order formulas. In *Proceedings of the 19th International Colloquium on Automata, Languages and Programming*, ICALP, pages 16–27.

Springer-Verlag, 1992.

[Str94] Howard Straubing. *Finite automata, formal logic, and circuit complexity*. Birkhäuser Boston Inc., 1994.

[STT95] Howard Straubing, Denis Thérien, and Wolfgang Thomas. Regular languages defined with generalized quantifiers. *Information and Computation*, 118(2):289–301, 1995.

[Tsa11] Todor Tsankov. The additive group of the rationals does not have an automatic presentation. *Journal of Symbolic Logic*, 76(4):1341–1351, 2011.