# Finitary Languages[⋆,⋆⋆]

Krishnendu Chatterjee[1] and Nathanaël Fijalkow[1,2]

[1] IST Austria (Institute of Science and Technology, Austria)
krishnendu.chatterjee@ist.ac.at
[2] ÉNS Cachan (École Normale Supérieure de Cachan, France)
nathanael.fijalkow@gmail.com

**Abstract.** The class of $\omega$-regular languages provides a robust specification language in verification. Every $\omega$-regular condition can be decomposed into a safety part and a liveness part. The liveness part ensures that something good happens "eventually". Finitary liveness was proposed by Alur and Henzinger as a stronger formulation of liveness [2]. It requires that there exists an unknown, fixed bound $b$ such that something good happens within $b$ transitions. In this work we consider automata with finitary acceptance conditions defined by finitary Büchi, parity and Streett languages. We give the topological complexity of finitary acceptance conditions, and present a regular-expression characterization of the languages expressed by finitary automata. We provide a classification of finitary and classical automata with respect to the expressive power, and give optimal algorithms for classical decisions problems on finitary automata. We (a) show that the finitary languages are $\Sigma_2^0$-complete; (b) present a complete picture of the expressive power of various classes of automata with finitary and infinitary acceptance conditions; (c) show that the languages defined by finitary parity automata exactly characterize the star-free fragment of $\omega B$-regular languages [4]; and (d) show that emptiness is NLOGSPACE-complete and universality as well as language inclusion are PSPACE-complete for finitary automata.

## 1 Introduction

**Classical $\omega$-regular languages: strengths and weakness.** The widely studied class of $\omega$-regular languages provides a robust specification language for solving control and verification problems (see, *e.g*, [13, 14]). Every $\omega$-regular specification can be decomposed into a safety part and a liveness part [1]. The safety part ensures that the component will not do anything "bad" (such as violate an invariant) within any finite number of transitions. The liveness part ensures that the component will do something "good" (such as proceed, or respond, or terminate) in the long-run. Liveness can be violated only in the limit, by infinite sequences of transitions, as no bound is stipulated on when the good thing must happen. This infinitary, classical formulation of liveness has both strengths

⋆ The research was supported by Austrian NFN ARiSE.
⋆⋆ Fuller version with proofs available at [7].

and weaknesses. A main strength is robustness, and in particular, independence from the chosen granularity of transitions. Another main strength is simplicity, allowing liveness to serve as an abstraction for complicated safety conditions. For example, a component may always respond in a number of transitions that depends, in some complicated manner, on the exact size of the stimulus. Yet for correctness, we may be interested only that the component will respond "eventually". However, these strengths also point to a weakness of the classical definition of liveness: it can be satisfied by components that in practice are quite unsatisfactory because no bound can be put on their response time.

**Stronger notion of liveness.** For the weakness of the infinitary formulation of liveness, alternative and stronger formulations of liveness have been proposed. One of these is *finitary* liveness [2]: finitary liveness does not insist on a response within a known bound $b$ (*i.e*, every stimulus is followed by a response within $b$ transitions), but on response within some unknown bound (*i.e*, there exists $b$ such that every stimulus is followed by a response within $b$ transitions). Note that in the finitary case, the bound $b$ may be arbitrarily large, but the response time must not grow forever from one stimulus to the next. In this way, finitary liveness still maintains the robustness (independence of step granularity) and simplicity (abstraction of complicated safety) of traditional liveness, while removing unsatisfactory implementations.

**Finitary parity and Streett conditions.** The classical infinitary notion of fairness is given by the Streett condition: it consists in a set of $d$ pairs of requests and corresponding responses (grants) and requires that every request that appears infinitely often must be responded infinitely often. Its finitary counterpart, the finitary Streett condition, requires that there is a bound $b$ such that in the limit every request is responded within $b$ steps. As a special case, the classical infinitary parity condition gives an total order of urgency over the requests. It consists in a priority function and requires that the minimum priority visited infinitely often is even. Its finitary counterpart, the finitary parity condition, requires that there is a bound $b$ such that in the limit after every odd priority a lower even priority is visited within $b$ steps.

**Results on classical automata.** There are several robust results on the languages expressible by automata with infinitary Büchi, parity and Streett conditions, as follows: (a) *topological complexity:* it is known that Büchi conditions are $\Pi_2^0$-complete, whereas parity and Streett conditions lie in the boolean closure of $\Sigma_2^0$ and $\Pi_2^0$ [12]; (b) *automata expressive power:* non-deterministic automata with Büchi conditions have the same expressive power as deterministic and non-deterministic parity and Streett automata [9, 15]; and (c) *regular expression characterization:* the class of languages expressed by deterministic parity is exactly defined by $\omega$-regular expressions (see the handbook [16] for details).

**Our results.** For finitary languages, topological, automata-theoretic, regular-expression and decision problems studies were all missing. In this work we present results in the four directions, as follows:

1. *Topological complexity.* We show that finitary Büchi, parity and Streett conditions are $\Sigma_2^0$-complete.

2. *Automata expressive power.* We show that finitary automata are incomparable in expressive power with classical automata. As in the infinitary setting, we show that non-deterministic automata with finitary Büchi, parity and Streett conditions have the same expressive power, as well as deterministic finitary parity and Streett automata, which are strictly more expressive than deterministic finitary Büchi automata. However, in contrast to the infinitary case, for finitary parity condition, non-deterministic automata are strictly more expressive than the deterministic counterpart. As a by-product we derive boolean closure properties for finitary automata.
3. *Regular expression characterization.* We consider the characterization of finitary automata through an extension of $\omega$-regular languages defined as $\omega B$-regular languages (see [4]). We show that non-deterministic finitary Büchi automata express exactly the star-free fragment of $\omega B$-regular languages.
4. *Decision problems.* We show that emptiness is NLOGSPACE-complete and universality as well as language inclusion are PSPACE-complete for finitary automata.

**Related works.** The notion of finitary liveness was introduced in [2], and games with finitary objectives were studied in [8]. A generalization of $\omega$-regular languages that allows to express bound requirements is $\omega B$-regular languages, introduced in [4]. Variants were studied in [5] (see [3] for a survey); a topological characterization has been given in [11]. Our work along with topological and automata-theoretic studies of finitary languages, explores the relation between finitary languages and $\omega B$-regular expressions, rather than identifying a subclass of $\omega B$-regular expressions. We identify the exact subclass of $\omega B$-regular expressions that corresponds to non-deterministic finitary parity automata.

## 2 Definitions

### 2.1 Topological Complexity of Languages

Let $\Sigma$ be a finite set, called the alphabet. A word $w$ is a sequence of letters, which can be either finite or infinite, denoted $w_0 w_1 \dots$ where $w_0, w_1$ are letters. A language is a set of words: $L \subseteq \Sigma^*$ is a language over finite words and $L \subseteq \Sigma^\omega$ over infinite words. The concatenation of words is denoted $\cdot$, and naturally extended to languages.

**Cantor topology and Borel hierarchy.** Cantor topology on $\Sigma^\omega$ is given by *open* sets: a language is open if it can be described as $W \cdot \Sigma^\omega$ where $W \subseteq \Sigma^*$. Let $\Sigma_1^0$ denote the open sets and $\Pi_1^0$ denote the closed sets (a language is closed if its complement is open): they form the first level of the Borel hierarchy. Inductively, we define: $\Sigma_{i+1}^0$ is obtained as countable union of $\Pi_i^0$ sets; and $\Pi_{i+1}^0$ is obtained as countable intersection of $\Sigma_i^0$ sets. The higher a language is in the Borel hierarchy, the higher its topological complexity.

Since the above classes are closed under continuous preimage, we can define the notion of Wadge reduction [17]: $L$ reduces to $L'$, denoted by $L \preceq L'$, if there exists a continuous function $f : \Sigma^\omega \to \Sigma^\omega$ such $L = f^-(L')$, where $f^-(L')$ is the preimage of $L'$ by $f$. A language is hard with respect to a class if all languages of this class reduce to it. If it additionally belongs to this class, then it is complete.

**Classical liveness conditions.** We now consider three classes of languages that are widespread in verification and specification. They define liveness conditions, *i.e*, intuitively say that something good will happen "eventually". For an infinite word $w$, let $\mathrm{Inf}(w) \subseteq \Sigma$ denote the set of letters that appear infinitely often in $w$. The Büchi condition for a given $F \subseteq \Sigma$ is defined as follows:

$$\text{Büchi}(F) = \{w \mid \mathrm{Inf}(w) \cap F \neq \emptyset\}$$

*i.e*, the Büchi condition requires that some letter in $F$ appears infinitely often. The parity condition for a given priority function $p : \Sigma \to \mathbb{N}$, that maps letters to integers (representing priorities), is defined as follows:

$$\mathrm{Parity}(p) = \{w \mid \min(p(\mathrm{Inf}(w))) \text{ is even}\}$$

*i.e*, the parity condition requires that the lowest priority which appears infinitely often is even. The Streett condition for a given $(R, G) = (R_i, G_i)_{1 \leq i \leq d}$, where $R_i, G_i \subseteq \Sigma$ are request-grant pairs, is defined as follows:

$$\mathrm{Streett}(R, G) = \{w \mid \forall i, 1 \leq i \leq d, \mathrm{Inf}(w) \cap R_i \neq \emptyset \Rightarrow \mathrm{Inf}(w) \cap G_i \neq \emptyset\}$$

*i.e*, the Streett condition requires that for all requests $R_i$ that appear infinitely often, the corresponding grant $G_i$ also appears infinitely often.

The following theorem presents the topological complexity of the classical languages:

**Theorem 1 (Topological complexity of classical languages [12]).**

- *For all* $\emptyset \subset\neq F \subset\neq \Sigma$, *the language* Büchi$(F)$ *is* $\Pi^0_2$*-complete.*
- *The parity and Streett languages lie in the boolean closure of* $\Sigma^0_2$ *and* $\Pi^0_2$.

### 2.2 Finitary Languages

The finitary parity and Streett conditions have been defined in [8]. We give their definitions, and specialize them to finitary Büchi conditions. Let $(R, G) = (R_i, G_i)_{1 \leq i \leq d}$, where $R_i, G_i \subseteq \Sigma$, the definition for FinStreett$(R, G)$ uses distance sequence as follows:

$$\mathrm{dist}^j_k(w, (R, G)) = \begin{cases} 0 & w_k \notin R_j \\ \inf\{k' - k \mid k' \geq k, w_{k'} \in G_j\} & w_k \in R_j \end{cases}$$

*i.e*, given a position $k$ where $R_j$ is requested, $\mathrm{dist}^j_k(w, (R, G))$ is the waiting time (number of transitions) between the request $R_j$ and the corresponding grant $G_j$. Note that $\inf(\emptyset) = \infty$. Then $\mathrm{dist}_k(w, (R, G)) = \max\{\mathrm{dist}^j_k(w, p) \mid 1 \leq j \leq d\}$ and:

$$\mathrm{FinStreett}(R, G) = \{w \mid \limsup_k \mathrm{dist}_k(w, (R, G)) < \infty\}$$

*i.e*, the finitary Streett condition requires the supremum limit of the distance sequence to be bounded.

Since parity languages are a particular case of Streett languages, where $G_1 \subseteq R_1 \subseteq G_2 \subseteq R_2 \ldots$, the latter allows to define $\mathrm{FinParity}(p)$. The same applies to finitary Büchi languages, which are a particular case of finitary parity languages where the letters from the set $F$ have priority 0 and others have priority 1. We get the following definitions. Let $p : \Sigma \to \mathbb{N}$ a priority function, we define:

$$\mathrm{dist}_k(w, p) = \inf\{k' - k \mid k' \geq k, p(w_{k'}) \text{ is even and } p(w_{k'}) \leq p(w_k)\}$$

*i.e*, given a position $k$ where $p(w_k)$ is odd, $\mathrm{dist}_k(w, p)$ is the waiting time between the odd priority $p(w_k)$ and a lower even priority. Then $\mathrm{FinParity}(p) = \{w \mid \limsup_k \mathrm{dist}_k(w, p) < \infty\}$. We define similarly the finitary Büchi language: given $F \subseteq \Sigma$, let:

$$\mathrm{next}_k(w, F) = \inf\{k' - k \mid k' \geq k, w_{k'} \in F\}$$

*i.e*, $\mathrm{next}_k(w, F)$ is the waiting time before visiting a letter in $F$. Then

$$\mathrm{FinBüchi}(F) = \{w \mid \limsup_k \mathrm{next}_k(w, F) < \infty\}.$$

### 2.3 Automata, $\omega$-regular and Finitary Languages

**Definition 1.** *An automaton is a tuple $\mathcal{A} = (Q, \Sigma, Q_0, \delta, Acc)$, where $Q$ is a finite set of states, $\Sigma$ is the finite input alphabet, $Q_0 \subseteq Q$ is the set of initial states, $\delta \subseteq Q \times \Sigma \times Q$ is the transition relation and $Acc \subseteq Q^\omega$ is the acceptance condition.*

An automaton is deterministic if it has a single initial state and for every state and letter there is at most one transition. The transition relation of deterministic automata are described by functions $\delta : Q \times \Sigma \to Q$. An automaton is complete if for every state and letter there is a transition.

**Acceptance conditions.** We will consider various acceptance conditions for automata obtained from the last section by considering $Q$ as the alphabet. Automata with finitary acceptance conditions are referred as finitary automata; classical automata are those equipped with infinitary acceptance conditions.

**Notation 1** *We use a standard notation to denote the set of languages recognized by some class of automata. The first letter is either $N$ or $D$, where $N$ stands for "non-deterministic" and $D$ stands for "deterministic". The last letter refers to the acceptance condition: $B$ stands for "Büchi", $P$ stands for "parity" and $S$ stands for "Streett". The acceptance condition may be prefixed by $F$ for "finitary". For example, $NP$ denotes non-deterministic parity automata, and $DFS$ denotes deterministic finitary Streett automata. We have the following combination:*

$$\begin{Bmatrix} N \\ D \end{Bmatrix} \cdot \begin{Bmatrix} F \\ \varepsilon \end{Bmatrix} \cdot \begin{Bmatrix} B \\ P \\ S \end{Bmatrix}$$

We denote by $\mathbb{L}_\omega$ the set of $\omega$-regular languages ([6, 15, 9, 10]):

$$\mathbb{L}_\omega = NB = DP = NP = DS = NS.$$

## 3 Topological Complexity

In this section we define a finitary operator $\mathsf{UniCloOmg}$ that allows us to relate finitary languages to their infinitary counterparts; we then give their topological complexity.

**Union-closed-omega-regular operator on languages.** Given a language $L \subseteq \Sigma^\omega$, the language $\mathsf{UniCloOmg}(L) \subseteq \Sigma^\omega$ is the *union* of the languages $M$ that are subsets of $L$, *$\omega$-regular* and *closed*, *i.e*, $\mathsf{UniCloOmg}(L) = \bigcup\{M \mid M \subseteq L, M \in \Pi_1, M \in \mathbb{L}_\omega\}$.

**Proposition 1.** *For all languages $L \subseteq \Sigma^\omega$ we have $\mathsf{UniCloOmg}(L) \in \Sigma_2^0$.*

The following lemma shows that $\mathrm{FinStreett}(R, G)$ is obtained by applying the $\mathsf{UniCloOmg}$ operator to $\mathrm{Streett}(R, G)$.

**Lemma 1.** *For all $(R, G) = (R_i, G_i)_{1 \leq i \leq d}$, where $R_i, G_i \subseteq \Sigma$, we have*

$$\mathsf{UniCloOmg}(\mathrm{Streett}(R, G)) = \mathrm{FinStreett}(R, G).$$

**Corollary 1.** *The following assertions hold:*

- *For all $p : \Sigma \to \mathbb{N}$, we have $\mathsf{UniCloOmg}(\mathrm{Parity}(p)) = \mathrm{FinParity}(p)$;*
- *For all $F \subseteq \Sigma$, we have $\mathsf{UniCloOmg}(\mathrm{Büchi}(F)) = \mathrm{FinBüchi}(F)$.*

**Theorem 2 (Topological characterization of finitary languages).** *The finitary Büchi, finitary parity and finitary Streett languages are $\Sigma_2^0$-complete.*

*Proof.* We show that if $\emptyset \subsetneq F \subsetneq \Sigma$, then $\mathrm{FinBüchi}(F)$ is $\Sigma_2^0$-complete. It follows from Corollary 1 that $\mathrm{FinBüchi}(F) \in \Sigma_2^0$. We now show that $\mathrm{FinBüchi}(F)$ is $\Sigma_2^0$-hard. By Theorem 1 we have that $\mathrm{Büchi}(\Sigma \setminus F)$ is $\Pi_2^0$-complete, hence $\Sigma^\omega \setminus \mathrm{Büchi}(\Sigma \setminus F)$ is $\Sigma_2^0$-complete. We present a topological reduction to show that $\Sigma^\omega \setminus \mathrm{Büchi}(\Sigma \setminus F) \preceq \mathrm{FinBüchi}(F)$. Let $b : \Sigma^\omega \to \Sigma^\omega$ be the stuttering function defined as follows:

$$\begin{array}{rl} w &= w_0 \quad w_1 \quad \ldots \qquad w_n \qquad \ldots \\ b(w) &= w_0 \underbrace{w_1 w_1}_{2} \ldots \underbrace{w_n w_n \ldots w_n}_{2^n} \ldots \end{array}$$

The function $b$ is continuous. We can easily check that the following holds:

$$\mathrm{Inf}(w) \subseteq F \quad \text{iff} \quad \exists B \in \mathbb{N}, \exists n \in \mathbb{N}, \forall k \geq n, \mathrm{next}_k(b(w), F) \leq B.$$

Hence we get $\Sigma^\omega \setminus \mathrm{Büchi}(\Sigma \setminus F) \preceq \mathrm{FinBüchi}(F)$, so $\mathrm{FinBüchi}(F)$ is $\Sigma_2^0$-complete. From this we can deduce the two other claims. $\square$

## 4 Expressive Power of Finitary Automata

In this section we consider finitary automata, and compare their expressive power to classical automata. We then address the question of determinization. Deterministic finitary automata enjoy nice properties that allow to describe languages they recognize using the $\mathsf{UniCloOmg}$ operator. As a by-product we get boolean closure properties of finitary automata.

### 4.1 Comparison with Classical Automata

Finitary conditions allow to express bounds requirements:

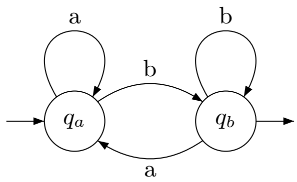


**Fig. 1.** A finitary Büchi automaton

*Example 1 (DFB $\not\subseteq \mathbb{L}_\omega$).* Consider the finitary Büchi automaton shown in Fig. 1, the state $q_b$ being its only final state. Its language is

$$L_B = \{(b^{j_0} a^{f(0)}) \cdot (b^{j_1} a^{f(1)}) \cdot (b^{j_2} a^{f(2)}) \ldots \mid f : \mathbb{N} \to \mathbb{N} \text{ bounded, and } j_i \in \mathbb{N}\}.$$

Indeed, reading the letter $b$ leads to the state $q_b$, and reading the letter $a$ leads to the state $q_a$. An infinite word is accepted if and only if the state $q_b$ is visited infinitely often and there is a bound between two consecutive visits. We can easily see that $L_B$ is not $\omega$-regular, using proof ideas from [4]: its complement would be $\omega$-regular, so it would contain ultimately periodic words, which is not the case.

However, finitary automata cannot distinguish between "many b's" and "only b's":

*Example 2 (DB $\not\subseteq$ NFB).* Consider the language of infinitely many $a$'s, *i.e*, $L_I = \{w \mid w \text{ has an infinite number of } a\}$, recognized by a simple deterministic Büchi automaton. However, we can show that there is no finitary Büchi automata that recognizes $L_I$. Intuitively, such an automaton would, while reading the infinite word $w = ab\ ab^2\ ab^3\ ab^4 \ldots ab^n \ldots \in L_I$, have to distinguish between all b's, otherwise it would accept a word with only b's at the end.

### 4.2 Deterministic Finitary Automata

Given a deterministic complete automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, Acc)$, we define its finitary restriction by $\mathsf{UniCloOmg}(\mathcal{A}) = (Q, \Sigma, q_0, \delta, \mathsf{UniCloOmg}(Acc))$.

Treating the automaton as a transducer, we consider the function $C_{\mathcal{A}} : \Sigma^\omega \to Q^\omega$ which maps an infinite word $w$ to the unique run $\rho$ of $\mathcal{A}$ on $w$ (there is a unique run since $\mathcal{A}$ is deterministic and complete). Then:

$$\mathcal{L}(\mathcal{A}) = \{w \mid C_{\mathcal{A}}(w) \in Acc\} = C_{\mathcal{A}}^{-}(Acc).$$

The property $C_{\mathcal{A}}^{-}(\mathsf{UniCloOmg}(Acc)) = \mathsf{UniCloOmg}(C_{\mathcal{A}}^{-}(Acc))$ is a direct consequence of the following lemma.

**Lemma 2.** *For all $\mathcal{A} = (Q, \Sigma, q_0, \delta, Acc)$ deterministic complete automaton, we have:*

1. *for all $A \subseteq Q^\omega$, if $A$ is closed then $C_{\mathcal{A}}^{-}(A)$ is closed ($C_{\mathcal{A}}$ is continuous).*
2. *for all $L \subseteq \Sigma^\omega$, if $L$ is closed then $C_{\mathcal{A}}(L)$ is closed ($C_{\mathcal{A}}$ is closed).*
3. *for all $A \subseteq Q^\omega$, if $A$ is $\omega$-regular then $C_{\mathcal{A}}^{-}(A)$ is $\omega$-regular.*
4. *for all $L \subseteq \Sigma^\omega$, if $L$ is $\omega$-regular then $C_{\mathcal{A}}(L)$ is $\omega$-regular.*

**Theorem 3.** *For any deterministic complete automaton $\mathcal{A}$ recognizing a language $L$, the finitary restriction of this automaton* $\mathsf{UniCloOmg}(\mathcal{A})$ *recognizes* $\mathsf{UniCloOmg}(L)$.

Theorem 3 allows to extend all known results on infinitary deterministic classes to finitary deterministic classes: as a corollary, we have $DFB \subsetneq DFP$ and $DFP = DFS$.

We now show that non-deterministic finitary parity automata are more expressive than deterministic finitary parity automata. However, for every language $L \in \mathbb{L}_\omega$ there exists $\mathcal{A} \in DP$ such that $\mathcal{A}$ recognizes $L$, and by Theorem 3 the deterministic finitary parity automaton $\mathsf{UniCloOmg}(\mathcal{A})$ recognizes $\mathsf{UniCloOmg}(L)$. Observe that Theorem 3 does not hold for non-deterministic automata, since we have $DP = NP$ but $DFP \subsetneq NFP$.

*Example 3 ($DFP \subsetneq NFP$).* As for Example 1 we consider the languages $L_1 = \{(a^{j_0} b^{f(0)}) \cdot (a^{j_1} b^{f(1)}) \cdot (a^{j_2} b^{f(2)}) \dots \mid f : \mathbb{N} \to \mathbb{N}, f \text{ bounded}, \forall i \in \mathbb{N}, j_i \in \mathbb{N}\}$ and $L_2 = \{(a^{f(0)} b^{j_0}) \cdot (a^{f(1)} b^{j_1}) \cdot (a^{f(2)} b^{j_2}) \dots \mid f : \mathbb{N} \to \mathbb{N}, f \text{ bounded}, \forall i \in \mathbb{N}, j_i \in \mathbb{N}\}$. It follows from Example 1 that both $L_1$ and $L_2$ belong to $DFP$, hence to $NFP$. A finitary parity automaton, relying on non-determinism, is easily built to recognize $L = L_1 \cup L_2$, hence $L \in NFP$. We can show that we cannot bypass this non-determinism, as by reading a word we have to decide well in advance which sequence will be bounded: a's or b's, *i.e*, $L \notin DFP$. To prove it, we interleave words of the form $(a^* \cdot b^*)^* \cdot a^\omega$ and $(a^* \cdot b^*)^* \cdot b^\omega$, and use a pumping argument to reach a contradiction.

### 4.3 Non-deterministic Finitary Automata

We can show that non-deterministic finitary Streett automata can be reduced to non-deterministic finitary Büchi automata, and this completes the picture of expressive power comparison.

Our results are summarized in Corollary 2 and shown in Fig 2.

**Corollary 2.** *We have (a) $DFB \not\subseteq \mathbb{L}_\omega$; (b) $DFB \subsetneq DFP = DFS \subsetneq NFB = NFP = NFS$; (c) $DB \not\subseteq NFB$; (d) $\mathbb{L}_\omega \not\subseteq NFB$.*

### 4.4 Closure Properties

**Theorem 4 (Closure properties).** *The following closure properties hold:*

1. *DFP is closed under intersection.*
2. *DFP is not closed under union.*
3. *NFP is closed under union and intersection.*
4. *DFP and NFP are not closed under complementation.*

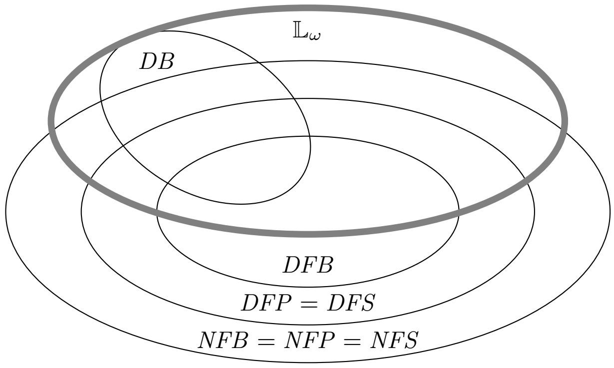


**Fig. 2.** Expressive power classification

## 5 Regular Expression Characterization

In this section we address the question of giving a syntactical representation of finitary languages, using a special class of regular expressions.

The class of $\omega B$-regular expressions was introduced in the work of [4] as an extension of $\omega$-regular expressions, as an attempt to express bounds in regular languages. To define $\omega B$-regular expressions, we need regular expressions and $\omega$-regular expressions.

Regular expressions define regular languages over finite words, and have the following grammar:

$$L := \emptyset \mid \varepsilon \mid \sigma \mid L \cdot L \mid L^* \mid L + L; \quad \sigma \in \Sigma$$

In the above grammar, $\cdot$ stands for concatenation, $*$ for Kleene star and $+$ for union. Then $\omega$-regular languages are finite unions of $L \cdot L'^{\omega}$, where $L$ and $L'$ are regular languages of finite words. The class of $\omega B$-regular languages, as defined in [4], is described by finite union of $L \cdot M^{\omega}$, where $L$ is a regular language over finite words and $M$ is a $B$-regular language over infinite sequences of finite words. The grammar for $B$-regular languages is as follows:

$$M := \emptyset \mid \varepsilon \mid \sigma \mid M \cdot M \mid M^* \mid M^B \mid M + M; \quad \sigma \in \Sigma$$

The semantics of regular languages over infinite sequences of finite words will assign to a $B$-regular expression $M$, a language in $(\Sigma^*)^{\omega}$. The infinite sequence $\langle u_0, u_1, \ldots \rangle$ will be denoted by $\boldsymbol{u}$. The semantics is defined by structural induction as follows.

- $\emptyset$ is the empty language,
- $\varepsilon$ is the language containing the single sequence $(\varepsilon, \varepsilon, \ldots)$,
- $a$ is the language containing the single sequence $(a, a, \ldots)$,
- $M_1 \cdot M_2$ is the language $\{\langle u_0 \cdot v_0, u_1 \cdot v_1, \ldots \rangle \mid \boldsymbol{u} \in M_1, \boldsymbol{v} \in M_2\}$,
- $M^*$ is the language $\{\langle u_0 \ldots u_{f(1)-1}, u_{f(1)} \ldots u_{f(2)-1}, \ldots \rangle \mid \boldsymbol{u} \in M, f : \mathbb{N} \to \mathbb{N}\}$,

- $M^B$ is defined like $M^*$ but we additionally require the values $f(i+1) - f(i)$ to be bounded uniformly in $i$,
- $M_1 + M_2$ is $\{\boldsymbol{w} \mid \boldsymbol{u} \in M_1, \boldsymbol{v} \in M_2, \forall i, w_i \in \{u_i, v_i\}\}$.

Finally, the $\omega$-operator on sequences with nonempty words on infinitely many coordinates is: $\langle u_0, u_1, \ldots \rangle^\omega = u_0 u_1 \ldots$. This operation is naturally extended to languages of sequences by taking the $\omega$ power of every sequence in the language. The class of $\omega B$-regular languages is more expressive than *NFB*, and this is due to the $*$-operator. We will consider the following fragment of $\omega B$-regular languages where we do not use the $*$-operator for $B$-regular expressions (however, the $*$-operator is allowed for $L$, regular languages over finite words). We call this fragment the star-free fragment of $\omega B$-regular languages.

For example, the language $L_B$ defined in Example 1, is described by the star-free $\omega B$-regular expression $(a^B \cdot b)^\omega$.

**Theorem 5.** *NFB exactly captures star-free $\omega B$-regular expressions.*

To prove that any language in *NFB* can be described by a star-free $\omega B$-regular expression, we use the same lines as for $\omega$-regular languages, except that a special attention is needed on size of final loops. The converse implication is more involved. We define acceptance conditions for automata reading infinite sequence of finite words, and proceed by induction on star-free $B$-regular expressions $M$ to build a finitary Büchi automaton that recognizes $M^B$. Then, we lift up automata reading infinite sequences of finite words to automata reading infinite words. This transformation is possible due to the *key*, yet simple observation that for all star-free $B$-regular expressions $M$ and for all $\boldsymbol{v} \in M$ we have that $(|v_n|)_n$ is bounded.

## 6 Decision Problems

In this section we consider the complexity of the decision problems for finitary languages. We present the results for finitary Büchi automata for simplicity, but the arguments for finitary parity and Streett automata are similar.

**Theorem 6 (Decision problems).** *The following assertions hold:*

1. (Emptiness). *Given a finitary Büchi automaton $\mathcal{A}$, whether $\mathcal{L}(\mathcal{A}) = \emptyset$ is* NLOGSPACE-*complete and can be decided in linear time.*
2. (Universality). *Given a finitary Büchi automaton $\mathcal{A}$ whether $\mathcal{L}(\mathcal{A}) = \Sigma^\omega$ is* PSPACE-*complete.*
3. (Language inclusion). *Given two finitary Büchi automata $\mathcal{A}$ and $\mathcal{B}$, whether $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$ is* PSPACE-*complete.*

We can show that a finitary Büchi automaton is empty if and only if it is empty regarded as a Büchi automaton. The PSPACE-hardness for universality and language inclusion follows from the special case of automata over finite words. For the PSPACE membership, we design a PSPACE algorithm for language inclusion (and universality follows as a special case), by performing a synchronous product of $\mathcal{A}$ and a subset construction of $\mathcal{B}$.

## References


1. Alpern, B., Schneider, F.B.: Defining Liveness. Information Processing Letters **21**(4) (1985) 181–185
2. Alur, R., Henzinger, T.A.: Finitary Fairness. ACM Transactions on Programming Languages and Systems **20**(6) (1998) 1171–1194
3. Bojańczyk, M.: Beyond omega-regular languages. In: International Symposium on Theoretical Aspects of Computer Science, STACS'10. (2010) 11–16
4. Bojańczyk, M., Colcombet, T.: Bounds in $\omega$-Regularity. In: Proceedings of the 21st Annual IEEE Symposium on Logic in Computer Science, LICS'06, IEEE Computer Society (2006) 285–296
5. Bojańczyk, M., Toruńczyk, S.: Deterministic Automata and Extensions of Weak MSO. In: International Conference on the Foundations of Software Technology and Theoretical Computer Science, FSTTCS'09. (2009) 73–84
6. Büchi, J.R.: On a decision method in restricted second-order arithmetic. In: Proceedings of the 1st International Congress of Logic, Methodology, and Philosophy of Science, CLMPS'60, Stanford University Press (1962) 1–11
7. Chatterjee, K., Fijalkow, N.: Finitary languages. CoRR **abs/1101.1727** (2011)
8. Chatterjee, K., Henzinger, T.A., Horn, F.: Finitary Winning in $\omega$-regular Games. ACM Transactions on Computational Logic **11**(1) (2009)
9. Choueka, Y.: Theories of automata on $\omega$-tapes: A simplified approach. Journal of Computer and System Sciences **8** (1974) 117–141
10. Gurevich, Y., Harrington, L.: Trees, Automata, and Games. In: Proceedings of the 14th Annual ACM Symposium on Theory of Computing, STOC'82, ACM Press (1982) 60–65
11. Hummel, S., Skrzypczak, M., Toruńczyk, S.: On the Topological Complexity of MSO+U and Related Automata Models. In: International Symposium on Mathematical Foundations of Computer Science, MFCS'10. (2010) 429–440
12. Manna, Z., Pnueli, A.: The Temporal Logic of Reactive and Concurrent Systems: Specification. Springer-Verlag (1992)
13. Pnueli, A., Rosner, R.: On the Synthesis of a Reactive Module. In: Proceedings of the 16th Annual ACM Symposium on Principles of Programming Languages, POPL'89. (1989) 179–190
14. Ramadge, P.J., Wonham, W.M.: Supervisory control of a class of discrete-event processes. SIAM Journal on Control and Optimization **25**(1) (1987) 206–230
15. Safra, S.: Exponential Determinization for $\omega$-Automata with Strong-Fairness Acceptance Condition. In: Annual ACM Symposium on Theory of Computing, STOC'92, ACM Press (1992)
16. Thomas, W.: Languages, Automata, and Logic. In Rozenberg, G., Salomaa, A., eds.: Handbook of Formal Languages. Volume 3, Beyond Words. Springer (1997) 389–455
17. Wadge, W.W.: Reducibility and Determinateness of Baire Spaces. PhD thesis, UC Berkeley (1984)