On the Complexity of Multi-entry DFAs

Nelma Moreira Rogério Reis Gonçalo Teixeira*

CMUP & DCC, Faculdade de Ciências da Universidade do Porto,
Rua do Campo Alegre, 4169-007 Porto, Portugal
{nelma.moreira, rogerio.reis, goncalo.teixeira}@fc.up.pt

Abstract

Multi-entry finite automata (MDFAs) are a generalization of DFAs that allow for an arbitrary number of initial states. This paper extends existing research on MDFAs by studying their operational state complexity and the computational complexity of their associated decision problems. In particular, we analyze the cost on the number of states of the standard language-theoretic operations when performed on MDFAs and compare these results with the well known complexities for DFAs and NFAs. Additionally, we also analyze the complexity of deciding the membership, emptiness, universality and inclusion problems for MDFAs. Our findings contribute to a deeper understanding of the role that nondeterminism plays in the computational difficulty of certain problems.

Keywords: Limited Nondeterminism, Operational State Complexity, Automata Theory.

1 Introduction

As the saying goes, "with great power there must also come great responsibility" [21], and nondeterminism is no exception. While NFAs can be exponentially more succinct than their deterministic counterparts, this expressive

*This work was partially supported by CMUP, which is financed by national funds through FCT under the project with DOI 10.54499/UID/00144/2025; and by AVATAR through the FCT project 2023.12169.PEX. And also project URBAN-NET, with the reference NORTE2030-FEDER-02697300, co-financed by the European Union, through the NORTE 2030 Regional Program, of Portugal 2030.

power comes at the cost of rendering most decision problems intractable. This tension has motivated the study of automata models in which only limited forms of nondeterminism are permitted.

Among the most natural of such restrictions is limiting nondeterminism to the choice of initial states, giving rise to *Multi-entry Finite Automata* (MDFA). In this model, the transition function remains fully deterministic, while the computation may begin in any one of the initial states. This model was first studied by Gill and Kou [8], who considered the special case in which every state is initial. Unfortunately, this restriction is too strong. The fact that this restricts the acceptable languages to a proper subclass of the regular languages, motivated Kappes [19] and Holzer et al. [12] to investigate the general case.

Despite their highly restricted form of nondeterminism, MDFAs already exhibit several properties that are present in NFAs. Kappes and Holzer et al. demonstrated that MDFAs can be exponentially more succinct than DFAs, via a modified powerset construction. Holzer et al. further established that MDFA minimization is PSPACE-COMPLETE. This was subsequently strengthened by Malcher [24], who showed that even if the number of initial states is fixed finding a minimal MDFA is still NP-COMPLETE.

What makes MDFAs particularly appealing is that, since they can be seen as a collection of identical DFAs that only differ in their initial states, they are exceptionally well suited for parallel and distributed implementation [8, 3, 4]. Despite this practical appeal, the theory of MDFAs remains largely unexplored.

In this paper, we address this gap by investigating the operational and computational complexities of MDFAs. Section 2 fixes the definitions and notation used throughout the rest of the paper. Section 2.1 formally introduces MDFAs. Section 3 analyzes the operational state complexity of standard language-theoretic operations on MDFAs. Finally, Section 4 establishes the computational complexity of deciding the emptiness, universality, and inclusion problems for MDFAs. All algorithms and constructions have been implemented in the FAdo [27] system and are available at https://codeberg.org/tttardigrado/mdfa-fado.

2 Preliminaries

We assume that the reader is familiar with the basic notions of automata and formal language theory. For details, we refer the reader to [13, 32].

Let Σ be a finite non-empty *alphabet* of symbols. A *word* w is a sequence of symbols of Σ. A *language* L is a set of words. The set of all words over Σ is $\Sigma^\star$. The *length of a word* $w \in \Sigma^\star$ is $|w|$, the number of occurrences of $a \in \Sigma$ in $w \in \Sigma^\star$ is $|w|_a$, and ε is the *empty word*. Given two integers $i, j \in \mathbb{Z}$, let $[i, j] := \{k \in \mathbb{Z} \mid i \leq k \leq j\}$, and $[j] := [0, i]$. The *cardinality of a set* S is $|S|$, and its *powerset* is 2^S.

A *nondeterministic finite automaton* (NFA) is a 5-tuple $N = \langle Q, \Sigma, \delta, I, F \rangle$, where Q is a finite set of states, Σ is a finite alphabet, $\delta : Q \times \Sigma \to 2^Q$ is the transition relation, $I \subseteq Q$ is the set of initial states, and $F \subseteq Q$ is the set of final states. The function δ is extended to $2^Q \times \Sigma^\star$ in the natural way. The language accepted by N is $\mathcal{L}(N) = \{w \in \Sigma^\star \mid \delta(I, w) \cap F \neq \emptyset\}$. Two NFAs are equivalent if they accept the same language.

A state q is said to be *reachable* (resp. *co-reachable*) if there is a word $w \in \Sigma^\star$ such that $q \in \delta(I, w)$ (resp. $\delta(q, w) \cap F \neq \emptyset$). An NFA is *reachable* (resp. *co-reachable*) if all of its states are reachable (resp. co-reachable). If an NFA is both reachable and co-reachable, then it is *trim*. The *reversal* of N is the NFA $N^R = \langle Q, \Sigma, \delta^R, F, I \rangle$, where $\delta^R(q, \sigma) = \{p \mid q \in \delta(p, \sigma)\}$, obtained by reversing the transitions, and swapping initial and final states.

The NFA N is *deterministic* (DFA) if it has a single initial state, i.e. $|I| = 1$, and all transitions are deterministic, i.e. $|\delta(q, \sigma)| \leq 1$, for all $q \in Q$ and $\sigma \in \Sigma$. Note that our definition allows for DFAs with a partial transition function. An NFA N can be converted into an equivalent DFA $\mathcal{D}(N) = \langle 2^Q, \Sigma, \delta, I, \{S \mid S \cap F \neq \emptyset\} \rangle$ by using the powerset construction.

A DFA without initial or final states $A = \langle Q, \Sigma, \delta \rangle$ is called a *semiautomaton* [29]. The semigroup $\mathcal{T}(A)$ generated by the set $\{\delta(-, \sigma) \mid \sigma \in \Sigma\}$ and with composition is said to be the *transition semigroup* induced by A. If the transition semigroup is isomorphic to the full transformation semigroup on Q, we say that it is *functionally complete*.

An NFA is *minimal* if it has the minimal number of states of all equivalent NFAs. The *nondeterministic state complexity* of a language L, $\mathrm{nsc}(L)$, is the number of states of its minimal NFA (normally considering $|I| = 1$). The operational state complexity of a k-ary operation $\otimes$ is the maximal nondeterministic state complexity of the languages $\otimes(L_1, \cdots, L_k)$ as a function of the (nondeterministic state) complexities of $L_1, \cdots, L_k$.

To prove minimality of NFAs, we use fooling sets [1, 2, 9]. A set of pairs of words $\mathcal{F} = \{\langle x_i, y_i \rangle \mid i \in [1, n]\}$, for $n \geq 1$, is called a *fooling set* for a language $L \subseteq \Sigma^\star$ if for each $i, j \in [1, n]$:

$$x_i y_i \in L \tag{F1}$$

$$i \neq j \implies x_i y_j \notin L \vee x_j y_i \notin L \tag{F2}$$

Lemma 2.1 (Birget [1]). *Let $\mathcal{F}$ be a fooling set for a regular language L. Then every* NFA *for L has at least $|\mathcal{F}|$ states.*

2.1 Multi-entry Finite Automata

The definition of DFAs as a restriction of NFAs imposing both a deterministic transition function and a single initial state naturally raises the question of what would be the properties of the models that arise by lifting only one of these constraints. This motivates the following definition:

A *Multi-entry finite automaton* (MDFA) is a 5-tuple $M = \langle Q, \Sigma, \delta, I, F\rangle$, where Q is a finite set of states, Σ is a finite alphabet, $\delta : Q \times \Sigma \to Q$ is a deterministic transition function, $I \subseteq Q$ is the set of initial states, and $F \subseteq Q$ is the set of final states.

The language accepted by an MDFA M is $\mathcal{L}(M) = \{w \in \Sigma^\star \mid \delta(I, w) \cap F \neq \emptyset\}$. The notions of equivalence, and minimality for MDFAs are defined analogously to those for NFAs.

3 Multi-entry Operational State Complexity

In this section we explore the operational state complexity of MDFAs. Following the definitions of Section 2, the *multi-entry state complexity* of a regular language L, $\mathrm{msc}(L)$, is defined as the number of states of a minimal MDFA recognizing L. The *multi-entry operational state complexity* of an operation is defined analogously to the nondeterministic case.

Since MDFAs are a special case of NFAs, Lemma 2.1 guarantees that fooling sets are an effective method to prove the minimality of MDFAs, as exemplified by the following lemma:

Lemma 3.1. *Let $n \geq 2$. Let $L_{na} = \{w \in \{a, b\}^\star \mid |w|_a \in [n]\}$. Then,* $\mathrm{msc}(L_{na}) = n + 1$.

Proof. Clearly, L_{na} is recognized by the $(n+1)$-state automaton of Fig. 1. Consider the following set of $n + 1$ pairs:

$$\mathcal{A} = \{\langle a^i, a^{n-i}\rangle \mid i \in [n]\}.$$

We will prove that $\mathcal{A}$ is a fooling set for L_{na}: (F1) For any i, the word $a^i a^{n-i}$ satisfies $|a^i a^{n-i}|_a = n$. Therefore, it belongs to L_{na}. (F2) If $j < i$, then $|a^i a^{n-j}|_a = n + i - j > n$ and the word does not belong to L_{na}. This shows that $\mathcal{A}$ is a fooling set for L_{na}, which proves the result. □

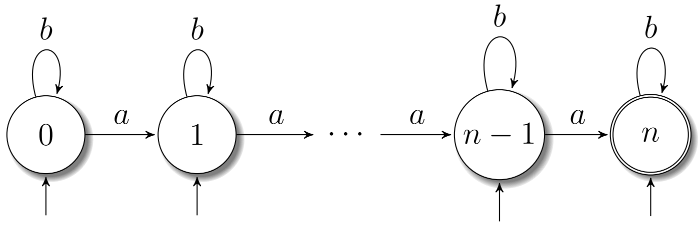

Figure 1: $(n+1)$-state MDFA recognizing L_{na}

3.1 Intersection and Union

Let us start by considering the *intersection* operation. It is well known [33, 11] that both the deterministic and nondeterministic state complexities of intersection achieve the same value of mn, where m and n are the number of states of the automata of the operands. This suggests that the multi-entry state complexity should also be the same, and Theorem 3.1 proves this claim.

Lemma 3.2. *Let $L_1, L_2 \subseteq \Sigma^\star$ be two languages with $\mathrm{msc}(L_1) \leq n$ and $\mathrm{msc}(L_2) \leq m$, for $n, m \geq 1$. Then, nm states are sufficient for an* MDFA *to recognize $L_1 \cap L_2$.*

Proof. Let $M_1 = \langle Q_1, \Sigma, \delta_1, I_1, F_1 \rangle$ and $M_2 = \langle Q_2, \Sigma, \delta_2, I_2, F_2 \rangle$ be two MDFAs that recognize L_1 and L_2, respectively. We define a product automaton $M = \langle Q_1 \times Q_2, \Sigma, \delta, I_1 \times I_2, F_1 \times F_2 \rangle$, where

$$\delta(\langle p, q \rangle, \sigma) = \langle \delta_1(p, \sigma), \delta_2(q, \sigma) \rangle,$$

if both $\delta_1(p, \sigma)$ and $\delta_2(q, \sigma)$ are well-defined. Since both δ_1 and δ_2 are deterministic, δ is also deterministic.

It is easy to see that $\mathcal{L}(M) = L_1 \cap L_2$, since $\delta(\langle i_1, i_2 \rangle, w) \in F_1 \times F_2$ iff $\delta_1(i_1, w) \in F_1$ and $\delta_2(i_2, w) \in F_2$, for any word $w \in \Sigma^\star$ and initial states $i_1 \in I_1$ and $i_2 \in I_2$. This implies that the construction is correct, thus establishing an upper bound for the multi-entry state complexity of intersection. □

Lemma 3.3. *Let $n, m \geq 2$. Let $L_1, L_2 \subseteq \Sigma^\star$ be two languages with $\mathrm{msc}(L_1) \leq n$ and $\mathrm{msc}(L_2) \leq m$. Then, nm states are necessary, in the worst case, for an* MDFA *to recognize $L_1 \cap L_2$.*

Proof. To prove that the bound of Lemma 3.2 is tight, consider the languages $L_{na} = \{w \in \{a,b\}^\star \mid |w|_a \in [n]\}$ and $L_{mb} = \{w \in \{a,b\}^\star \mid |w|_b \in [m]\}$ of Lemma 3.1. We want to prove that the set of pairs

$$\mathcal{A} = \{\langle a^i b^j, a^{n-i} b^{m-j}\rangle \mid i \in [n], j \in [m]\}$$

is a fooling set for $L_{na} \cap L_{mb}$: (F1) For any $\langle i,j\rangle$, the word $a^i b^j a^{n-i} b^{m-j}$ contains n occurrences of a and m occurrences of b. Thus, the word belongs to $L_{na} \cap L_{mb}$. (F2) Let $\langle i,j\rangle \neq \langle k,\ell\rangle$: If $k < i$, then $|a^i b^j a^{n-k} b^{m-\ell}|_a = n + (i-k) > n$, so the word does not belong to $L_{na} \cap L_{mb}$. If $i = k$ and $\ell < j$, then $|a^i b^j a^{n-k} b^{m-\ell}|_b = m + (j-\ell) > m$, and the word is not in $L_{na} \cap L_{mb}$. Therefore, $\mathcal{A}$ is a fooling set of size $(m+1)(n+1)$ for $L_{na} \cap L_{mb}$, and any MDFA recognizing it requires at least $(m+1)(n+1)$ states. □

From Lemmas 3.2 to 3.3, we get the following theorem:

Theorem 3.1 (Intersection)**.** *Let $n, m \geq 2$. Let $L_1, L_2 \subseteq \Sigma^\star$ be two languages with $\operatorname{msc}(L_1) \leq n$ and $\operatorname{msc}(L_2) \leq m$, for $n, m \geq 1$. Then, $\operatorname{msc}(L_1 \cap L_2) \leq mn$, and this bound is tight for $|\Sigma| \geq 2$.*

The case for the *union* operation is slightly different. The deterministic state complexity achieves the same bound as intersection [33], but nondeterminism allows us to reduce this value to $n + m + 1$ in the case that we enforce a single initial state, and $n + m$ in the general case [12, 14]. This raises the following question: is initial state nondeterminism enough to lower this bound? Theorem 3.2 answers this question positively.

Lemma 3.4. *Let $L_1, L_2 \subseteq \Sigma^\star$ be two languages with $\operatorname{msc}(L_1) \leq n$ and $\operatorname{msc}(L_2) \leq m$. Then, $n + m$ states are sufficient for an MDFA to recognize $L_1 \cup L_2$.*

Proof. Let $M_1 = \langle Q_1, \Sigma, \delta_1, I_1, F_1\rangle$ and $M_2 = \langle Q_2, \Sigma, \delta_2, I_2, F_2\rangle$ be MDFAs that recognize L_1 and L_2, respectively. Without loss of generality, we assume Q_1 and Q_2 are disjoint. We define a union automaton $M = \langle Q_1 \cup Q_2, \Sigma, \delta, I_1 \cup I_2, F_1 \cup F_2\rangle$, where

$$\delta(q,\sigma) = \begin{cases} \delta_1(q,\sigma) & \text{if } q \in Q_1 \text{ and } \delta_1(q,\sigma) \text{ is well-defined} \\ \delta_2(q,\sigma) & \text{if } q \in Q_2 \text{ and } \delta_2(q,\sigma) \text{ is well-defined} \\ \perp & \text{otherwise} \end{cases}$$

To prove the upper bound, we want to show that $\mathcal{L}(M) = L_1 \cup L_2$.

($\Longrightarrow$) Let $w \in \mathcal{L}(M)$. By definition, $\delta(i,w) \in F_1 + F_2$ for some $i \in I_1 + I_2$. There are two possibilities:

1. If $i \in I_1$, then $\delta(i, w) \in F_1$, so $w \in L_1$,
2. If $i \in I_2$, then $\delta(i, w) \in F_2$, so $w \in L_2$.

Hence, $w \in L_1 \cup L_2$.

($\Longleftarrow$) Let $w \in L_1 \cup L_2$. If $w \in L_1$, then there exists an initial state $i \in I_1$ such that $\delta_1(i, w) \in F_1$. Similarly, if $w \in L_2$, then there exists an initial state $i \in I_2$ such that $\delta_2(i, w) \in F_2$. In either case, we have that $\delta(i, w) \in F_1 + F_2$ for some $i \in I_1 + I_2$, and thus $w \in \mathcal{L}(M)$. □

Lemma 3.5. *Let $n, m \geq 2$. Let $L_1, L_2 \subseteq \Sigma^\star$ be two languages with $\operatorname{msc}(L_1) \leq n$ and $\operatorname{msc}(L_2) \leq m$. Then, $n + m$ states are necessary, in the worst case, for an* MDFA *to recognize $L_1 \cup L_2$.*

Proof. Consider again the witness families L_{na} and L_{mb} used in the proof of Lemma 3.3. Let

$$\mathcal{A} = \{\langle a^i, a^{n-i}\rangle \mid i \in [1, n]\} \cup \{\langle a^n, a^n\rangle\}$$
$$\mathcal{B} = \{\langle b^i, b^{m-i}\rangle \mid i \in [1, m]\} \cup \{\langle b^m, b^m\rangle\}$$

be two fooling sets of size $n + 1$ and $m + 1$ for L_{na} and L_{mb}, respectively. Since $\mathcal{A}$ and $\mathcal{B}$ are disjoint, their union $\mathcal{A} \cup \mathcal{B}$ is a fooling set of size $(n + 1) + (m + 1)$ for $L_{na} \cup L_{mb}$. By Lemma 2.1, we conclude that the minimal MDFA recognizing $L_{na} \cup L_{mb}$ has, at least, $(n + 1) + (m + 1)$ states, thus proving that the bound is tight. □

Lemmas 3.4 to 3.5 establish the following result:

Theorem 3.2 (Union)**.** *Let $n, m \geq 2$. Let $L_1, L_2 \subseteq \Sigma^\star$ be two languages with $\operatorname{msc}(L_1) \leq n$ and $\operatorname{msc}(L_2) \leq m$. Then, $\operatorname{msc}(L_1 \cup L_2) \leq m + n$, and this bound is tight for $|\Sigma| \geq 2$.*

3.2 Complementation

The last Boolean operation we consider is *complementation*. This operation is an entirely different beast: contrary to the other two, it is easy on DFAs, requiring no increase in the number of states, but can be exponentially expensive for NFAs, requiring 2^n states in the worst case [16]. This raises the following question: is this exponential blow-up an inherent consequence of any form of nondeterminism, or is nondeterminism in the *transition function* itself necessary to achieve this bound? We begin by establishing the same upper bound as in the general case for NFAs.

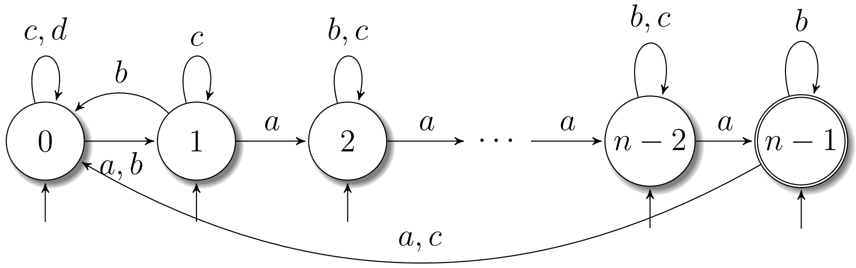

Figure 2: Variation on Brzozowski's Universal Witness automaton $\mathcal{M}_n$

Lemma 3.6. *Let* $L \subseteq \Sigma^\star$ *be a language with* $\mathrm{msc}(L) \leq n$*. Then,* 2^n *states are sufficient for an* MDFA *to recognize* $\overline{L}$*.*

Proof. Let M be an n-state MDFA recognizing L. By applying the powerset construction to M, we obtain an equivalent DFA $\mathcal{D}(M)$ with at most $2^n - 1$ states. Complementation of an (incomplete) DFA is achieved by adding a sink state and swapping final and non-final states. The resulting DFA $\overline{\mathcal{D}(M)}$ therefore recognizes $\overline{L}$ and has at most 2^n states. $\square$

We now show that this bound is tight. Let $n \geq 3$. Consider the automata $\mathcal{M}_n = \langle Q, \Sigma, \delta, Q, F \rangle$, of Fig. 2, where for $n \geq 3$, $Q = [n-1]$, $\Sigma = \{a, b, c, d\}$, $F = \{n-1\}$, and

$$
\begin{aligned}
\delta(q,a) &= (q+1) \bmod n & & & (\forall q \in [n-1]),\\
\delta(0,b) &= 1, \quad \delta(1,b) = 0, & \delta(q,b) &= q & (\forall q \in [2, n-1]),\\
\delta(n-1,c) &= 0, \quad \delta(q,c) = q & & & (\forall q \in [0, n-2]),\\
\delta(0,d) &= 0.
\end{aligned}
$$

Let $\mathcal{M}'_n$ denote the restriction of $\mathcal{M}_n$ to the alphabet $\Sigma' = \{a, b, c\}$. From an algebraic perspective, the input symbols act as follows: a induces the cyclic shift $(0, 1, \ldots, n-1)$, b induces the transposition $(0, 1)$, and c induces the singular transformation $\binom{n-1}{0}$. Consequently, $\mathcal{M}'_n$ is a variation of Brzozowski's universal witness automaton [5] in which all states are initial. In particular, the associated semiautomaton coincides with that of the universal witnesses, and its transition semigroup $\mathcal{T}(\mathcal{M}'_n)$ is therefore *functionally complete* as a consequence of a result of Piccard [28] and Dénes [7].

To establish tightness, we use the following lemma due to Hospodár et al. [14, Lemma 5].

Lemma 3.7 (Hospodár et al.)**.** *Let N be an n-state* NFA *in which every subset of the state set is reachable as well as co-reachable. Then, every NFA for the complement of the language $\mathcal{L}(N)$ has at least 2^n states.*

In order to apply this lemma, we start by proving that all 2^n subsets of states of $\mathcal{M}_n$ are reachable, and then that they are co-reachable.

Lemma 3.8. *Consider the automaton $\mathcal{M}_n$. If a subset $S \subseteq Q$ is reachable, then all other subsets of the same cardinality are also reachable.*

Proof. Fix a reachable subset $S \subseteq Q$. Since S is reachable, there is a word $w \in \Sigma^\star$ such that $\delta(Q, w) = S$. Let $T \subseteq Q$ be another subset with the same cardinality. There is a Q permutation π that maps S to T. Since $\mathcal{M}_n$ is functionally complete, there is a word $v \in \Sigma^\star$ whose action on the set of states corresponds to π. Thus, $\delta(Q, wv) = T$, and T is reachable. □

Lemma 3.9. *Consider the automaton $\mathcal{M}_n$. If a subset $S \subseteq Q$ with $|S| \geq 2$ is reachable, then all subsets of cardinality $|S| - 1$ are also reachable.*

Proof. Fix a reachable subset $S \subseteq Q$ with at least two elements $p, q \in S$. Since S is reachable, there is a word $w \in \Sigma^\star$ such that $\delta(Q, w) = S$. There is a Q permutation π that maps p to 0, q to $n-1$. From functional completeness, we can establish the existence of a word $v \in \Sigma^\star$ whose action on the set of states corresponds to π. Since $c \in \Sigma$ maps both 0 and 1 to the same state, the set $T = \delta(Q, wvc)$ is reachable and has cardinality $|S| - 1$. Since a set of cardinality $|S| - 1$ is reachable, then, by Lemma 3.8, all other subsets of the same cardinality are reachable. □

Since all states are initial, the full set Q is reachable. By repeated application of Lemma 3.9, all non-empty subsets are reachable. Because $\mathcal{M}_n$ is incomplete, the empty set is also reachable, and we can state the follow Lemma:

Lemma 3.10. *All 2^n subsets of the state set of $\mathcal{M}_n$ are reachable.*

We now establish co-reachability.

Lemma 3.11. *All 2^n subsets of the state set of $\mathcal{M}_n$ are co-reachable.*

Proof. Salomaa et al. [30] showed that any automaton recognizing a non-empty, non-universal language over an alphabet with at least three symbols has 2^n co-reachable subsets of states, provided its associated semiautomaton is functionally complete. Since $\mathcal{M}'_n$ satisfies these conditions and co-reachability is independent of the choice of initial states and the addition of a transition by $d \in \Sigma$, all 2^n subsets of the state set of $\mathcal{M}_n$ are co-reachable. $\square$

Hospodár et al. [14, Lemma 2] showed that a language L admits a fooling set of size n iff there exists an NFA recognizing L in which every singleton set of states is both reachable and co-reachable. Since every subset of states of $\mathcal{M}_n$ is both reachable and co-reachable, it follows in particular that all singleton subsets satisfy this property. Hence, the language recognized by $\mathcal{M}_n$ admits a fooling set of size n. By Lemma 2.1, this implies the following Lemma:

Lemma 3.12. *The* MDFA $\mathcal{M}_n$ *is minimal.*

Moreover, since all 2^n subsets are both reachable and co-reachable, it follows from Lemma 3.7 that the bound is tight for MDFAs over an alphabet with at least 4 symbols.

Theorem 3.3 (Complementation)**.** *Let $n \geq 3$. Let $L \subseteq \Sigma^\star$ be a language with* $\operatorname{msc}(L) \leq n$*. Then,* $\operatorname{msc}(\overline{L}) \leq 2^n$*, and this bound is tight for $|\Sigma| \geq 4$.*

3.3 Reversal

We now turn to the reversal operation. It is well known [33, 26] that although reversal preserves the size of NFAs, it leads to an exponential blowup in the number of states for DFAs. We show that MDFAs exhibit the same exponential worst-case behavior as single initial state DFAs.

Lemma 3.13. *Let L be a language with* $\operatorname{msc}(L) \leq n$*. Then $2^n - 1$ states suffice for an* MDFA *to recognize L^R.*

Proof. Let $M = \langle Q, \Sigma, \delta, I, F \rangle$ be an n-state MDFA with $\mathcal{L}(M) = L$. Since $\mathcal{L}(\mathcal{D}(M^R)) = L^R$, and $|\mathcal{D}(M^R)| \leq 2^n - 1$ in the worst case, the upper bound follows. $\square$

To establish tightness, we use the *some-register-on* language R_n introduced by Leung [22, 23]. Consider n Boolean-valued registers, with register 1 initially set to 1. The alphabet consists of copy instructions Copy i to j, abbreviated as i_j, each copying the value of register i into register j. A word belongs to R_n iff, after executing the corresponding sequence of copy instructions, at least one register is 1.

Lemma 3.14 (Leung [23])**.** *There exists an n-state* MDFA *recognizing R_n^R.*

Proof. Let $N = \langle [1,n], \Sigma, \delta, \{1\}, [1,n] \rangle$, where $\Sigma = \{i_j \mid i \neq j\}$ and $\delta = \{\langle k, i_j, k\rangle \mid i \neq j, j \neq k\} \cup \{\langle i, i_j, j\rangle \mid i \neq j\}$. It's easy to see that $\mathcal{L}(N) = R_n$: state k is active after reading a word w if and only if the corresponding register is on. Moreover, N^R's transition function is deterministic, so R_n^R is recognized by an n-state MDFA. □

Lemma 3.15 (Leung [23]). *Exactly* $2^n - 1$ *states are both sufficient and necessary for an* MDFA *to recognize* R_n.

Proof. Let $M = \langle Q, \Sigma, \delta, \{10^{n-1}\}, Q\rangle$, where $Q = \{0,1\}^n \setminus \{0^n\}$ is the set of all nonzero binary strings of length n, $\Sigma = \{i_j \mid i \neq j\}$, and $\delta(q, i_j) = q[j \mapsto q[i]]$, with $q[j \mapsto q[i]]$ denoting the string obtained from q by replacing its j-th bit with its i-th bit. That $\mathcal{L}(M) = R_n$ and $|M| = 2^n - 1$ follow directly from the definition. The proof that this MDFA is minimal is rather lengthy, and we refer the reader to the original paper by Leung [23]. □

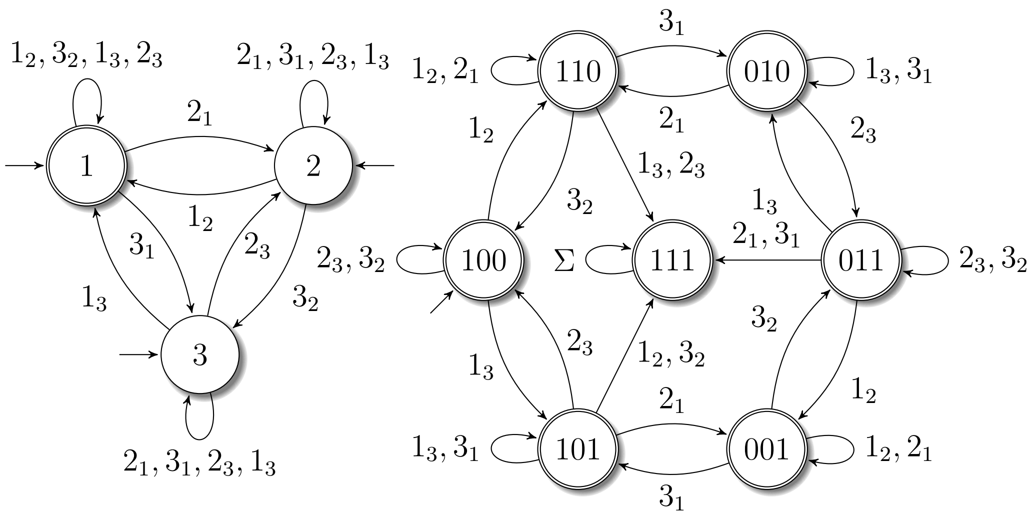

Figure 3: 3-state MDFA for R_3^R and $(2^3 - 1)$-state MDFA for R_3

Since the reversal operator is idempotent, i.e., $(R_n^R)^R = R_n$, the following theorem is an immediate consequence of Lemmas 3.13 to 3.15.

Theorem 3.4 (Reversal). *Let* $L \subseteq \Sigma^\star$ *be a language with* $\mathrm{msc}(L) \leq n$. *Then,* $\mathrm{msc}(L^R) \leq 2^n - 1$, *and this bound is tight for* $|\Sigma| = \mathcal{O}(n^2)$.

3.4 Closures Operations

The state complexities of the *suffix*, *prefix*, and *factor* closure operations have been investigated in detail by Brzozowski et al. [6] and Kao et al. [18]. They established that, in the worst case, the prefix, suffix, and factor closure operations require $2^n - 1$, n, and 2^{n-1} states, respectively, when applied to an n-state DFA. In this section we show that initial state nondeterminism alone is sufficient to reduce all these bounds to n.

Let M be an MDFA accepting L. By making all reachable states initial, we get an MDFA that accepts $\mathrm{Suff}(L)$. By making all co-reachable states final, we get an MDFA that accepts $\mathrm{Pref}(L)$. Finally, by applying both of the previous constructions we get an MDFA that acceptes $\mathrm{Fact}(L)$. This allows us to state the following upper bounds.

Lemma 3.16. *Let $L \subseteq \Sigma^\star$ be a language with* $\mathrm{msc}(L) \leq n$*. Then, n states are sufficient for an* MDFA *to recognize* $\mathrm{Suff}(L)$, $\mathrm{Pref}(L)$*, and* $\mathrm{Fact}(L)$*.*

Lemma 3.17. *Let $L \subseteq \Sigma^\star$ be a language with* $\mathrm{msc}(L) \leq n$*. Then, n states are sufficient for an* MDFA *to recognize* $\mathrm{Suff}(L)$*.*

Proof. Let $M = \langle Q, \Sigma, \delta, I, F\rangle$ be an n-state MDFA that recognizes L. We define $M_S = \langle Q, \Sigma, \delta, I', F\rangle$ to be the automaton constructed from M by making every reachable state initial. Let us start by proving that $\mathcal{L}(M_S) = \mathrm{Suff}(L)$.

($\Longrightarrow$) Let $w \in \mathcal{L}(M_S)$. Then, there exists an initial state $i' \in I'$ such that $\delta(i', w) \in F$. Since every initial state is reachable in M, there is a word $u \in \Sigma^\star$ such that $\delta(i, u) = i'$, thus $\delta(i, uw) \in F$. This means that $uw \in L$, and so $w \in \mathrm{Suff}(L)$.

($\Longleftarrow$) Conversely, let $w \in \mathrm{Suff}(L)$. By definition, there exists a word $u \in \Sigma^\star$ such that $uw \in L$, i.e., $\delta(i, uw) \in F$ for some initial state $i \in I$. Since all reachable state in M are initial in M_S, $\delta(i, u) \in I'$, thus proving that $w \in \mathcal{L}(M)$.

$\square$

Lemma 3.18. *Let $L \subseteq \Sigma^\star$ be a language with* $\mathrm{msc}(L) \leq n$*. Then, n states are sufficient for an* MDFA *to recognize* $\mathrm{Pref}(L)$*.*

Proof. Let $M = \langle Q, \Sigma, \delta, I, F\rangle$ be an n-state MDFA that recognizes L. We define $M_P = \langle Q, \Sigma, \delta, I, F'\rangle$ to be the automaton constructed from M by making every co-reachable state final. Let us start by proving that $\mathcal{L}(M_P) = \text{Pref}(L)$.

($\Longrightarrow$) Let $w \in \mathcal{L}(M_P)$. By definition, there is an initial state $i \in I$ such that $\delta(i, w) \in F'$. Since every final state of M_S is co-reachable in M, there is a word $u \in \Sigma^\star$ such that $\delta(i, wu) \in F$. This means that $wu \in L$ and, therefore, that $w \in \text{Pref}(L)$.

($\Longleftarrow$) Conversely, let $w \in \text{Pref}(L)$. This means that there exists a word $u \in \Sigma^\star$ such that $wu \in L$, i.e., $\delta(i, wu) \in F$ for some initial state $i \in I$. Since all co-reachable states in M are final in M_S, $\delta(i, w) \in F'$, thus proving that $w \in \mathcal{L}(M)$. $\square$

Lemma 3.19. *Let $L \subseteq \Sigma^\star$ be a language with* $\text{msc}(L) \leq n$*. Then, n states are sufficient for an* MDFA *to recognize* $\text{Fact}(L)$.

Proof. Let $M = \langle Q, \Sigma, \delta, I, F \rangle$ be an n-state MDFA that recognizes L. We define $M_F = \langle Q, \Sigma, \delta, I', F' \rangle$ to be the automaton constructed from M by making every reachable state initial and every co-reachable state final. Let us start by proving that $\mathcal{L}(M_F) = \text{Fact}(L)$.

($\Longrightarrow$) Let $w \in \mathcal{L}(M_F)$. By definition, there is an initial state $i' \in I'$ such that $\delta(i, w) \in F'$. Since every initial state of M_F is reachable in M and every final state of M_F is co-reachable in M, there are words $u, v \in \Sigma^\star$ such that $\delta(i, uwv) \in F$ for some initial state $i \in I$. This means that $uwv \in L$ and, therefore, that $w \in \text{Fact}(L)$.

($\Longleftarrow$) Conversely, let $w \in \text{Fact}(L)$. This means that there are words $u, v \in \Sigma^\star$ such that $uwv \in L$, i.e., $\delta(i, uwv) \in F$ for some initial state $i \in I$. Since all reachable states in M are initial in M_F, $\delta(i, u) \in I'$. Similarly, since all co-reachable states in M are final in M_F, $\delta(i, uw) \in F'$, thus proving that $w \in \mathcal{L}(M)$. $\square$

The following lemma proves the tightness of the three upper bounds:

Lemma 3.20. *Let $n \geq 2$. Let $L \subseteq \Sigma^\star$ be a language with* $\text{msc}(L) \leq n$*. Then, n states are necessary, in the worst case, for an* MDFA *to recognize* $\text{Suff}(L)$*,* $\text{Pref}(L)$*, and* $\text{Fact}(L)$.

Proof. Let $w \in \Sigma^\star$ be a word of length $n - 1$, and let M_w be the linear n-state MDFA recognizing the singleton language $\{w\}$. By the constructions of Lemma 3.16, we obtain n-state MDFAs M_S, M_P, and M_F recognizing $\text{Suff}(\{w\})$, $\text{Pref}(\{w\})$, and $\text{Fact}(\{w\})$, respectively.

We now prove that these automata are minimal. We start by noting that the four languages considered are finite and that the largest word that belongs to each of them is w. Suppose, for the sake of contradiction, that there are MDFAs recognizing these with less than n states. By the pigeonhole principle, any accepting computation of w must visit some states more than once. This implies the existence of a loop which we can iterate over and, thus, accept words of an arbitrary length. This contradicts finiteness and, thus, such an automaton cannot exist. Hence, the automata M_w, M_S, M_P, and M_F are minimal, and the bound of n states is tight. $\square$

From Lemmas 3.16 to 3.20, we can directly conclude that:

Theorem 3.5. *Let $n \geq 2$. Let $L \subseteq \Sigma^\star$ be a language with $\operatorname{msc}(L) \leq n$. Then, $\operatorname{msc}(\operatorname{Suff}(L)) \leq n$, $\operatorname{msc}(\operatorname{Pref}(L)) \leq n$ and $\operatorname{msc}(\operatorname{Fact}(L)) \leq n$, and these bounds are tight for $|\Sigma| \geq 2$.*

4 Computational Complexity

The computational complexity of decision problems for finite automata has been extensively studied [25, 10]. For DFAs, many classical decision problems have efficient algorithms, whereas for NFAs the presence of unrestricted nondeterminism often leads to many of those problems becoming computationally intractable in practice. Malcher [24] studied the complexities of the decision problems for MDFAs in which the number of initial states was given as part of the instance and proved that, other than minimization, all of them were in P. In this section we investigate the complexity of several decision problems for MDFAs in the general case.

We begin by considering the membership problem for MDFAs. In the classical case for DFAs, the membership problem is decidable in L, while for NFAs nondeterminism makes it NL-complete. We show that the limited kind of nondeterminism available to MDFAs does not suffice to make this problem computationally harder.

Definition 4.1 (MDFA-Membership)**.**
Input: An MDFA $M = \langle Q, \Sigma, \delta, I, F \rangle$ and a word $w \in \Sigma^\star$
Question: Does $w \in \mathcal{L}(M)$?

Theorem 4.2. MDFA-Membership *belongs to* L*.*

Proof. Let $M = \langle Q, \Sigma, \delta, I, F \rangle$ be an n-state MDFA and let $w \in \Sigma^\star$. We describe a deterministic logspace machine that decides whether $w \in \mathcal{L}(M)$.

For each initial state $i \in I$, we simulate running M with input w from that state. Since δ is deterministic, this can be done by simply keeping track of the current state. After processing w, the machine checks whether the resulting state is final and accepts iff this holds for some $i \in I$.

Since the algorithm only needs to store a counter for the initial states and the current state, which both can be stored using $\mathcal{O}(\log n)$ space, the total working space is $\mathcal{O}(\log n)$. Therefore, MDFA-MEMBERSHIP belongs to L. □

Next, we analyze the complexity of deciding whether the language recognized by an MDFA is empty. Contrary to the problems considered after this one, it follows from the fact that the emptiness problems for DFAs and NFAs are NL-COMPLETE that the MDFA version is also NL-COMPLETE.

Definition 4.3 (MDFA-EMPTINESS)**.**
Input: An MDFA $M = \langle Q, \Sigma, \delta, I, F \rangle$.
Question: Does $\mathcal{L}(M) = \emptyset$?

Theorem 4.4. MDFA-EMPTINESS *is* NL-COMPLETE.

Proof. Since, by the Immerman-Szelepcsényi theorem [15, 31], NL = CONL it suffices to show that the complementary problem is in NL. Let $M = \langle Q, \Sigma, \delta, I, F \rangle$ be an n-state MDFA. Notice that $\mathcal{L}(M) \neq \emptyset$ iff there exists a word labeling a path that is both initial and final. A nondeterministic logspace machine can verify this property as follows: it first guesses an initial state, then repeatedly guesses outgoing transitions, and accepts as soon as a final state is reached. To ensure termination, the machine rejects if more than n transitions are taken. This upper bound on the length of the accepting path is sound due to the fact that any accepting path of a longer length must contain a loop, and, thus, can be shortened by removing said loop. The algorithm stores only the current state and a counter, requiring only $\mathcal{O}(\log n)$ space. Hence, the problem is in NL.

Hardness follows from the fact that the specialization of this problem to DFAs is well known to be NL-COMPLETE [17]. Therefore, MDFA-EMPTINESS is NL-COMPLETE. □

We now turn to the universality problem, which asks whether a given automaton accepts all possible words over its alphabet. Contrary to the previous problem, initial state nondeterminism is already sufficient to make this problem computationally intractable.

Definition 4.5 (MDFA-Universality).
Input: An MDFA $M = \langle Q, \Sigma, \delta, I, F\rangle$.
Question: Does $\mathcal{L}(M) = \Sigma^\star$?

Theorem 4.6. MDFA-Universality *is* PSpace-complete.

Proof. Membership in PSpace follows from the fact that the general problem for NFAs is PSpace-complete [25].

Then, we prove hardness via a reduction from the DFA-Intersection-Emptiness problem (DIE), which is known to be PSpace-complete [20]. Let $\langle A_1, \cdots, A_k\rangle$ be an instance of the DIE problem. By De Morgan's laws, we have

$$\bigcap_{i=1}^{k} \mathcal{L}(M_i) = \emptyset \quad \text{iff} \quad \bigcup_{i=1}^{k} \overline{\mathcal{L}(M_i)} = \Sigma^\star.$$

The union of the complemented DFAs can be represented as an MDFA accepting the same language. Since complementation and union of DFAs can be performed in polynomial time on the size of the input, MDFA-Universality is PSpace-hard. This concludes the proof. $\square$

Finally, we consider the language inclusion problem for MDFAs which generalizes universality and is central in many verification tasks. Similarly to the universality problem, initial state nondeterminism makes this problem much harder.

Definition 4.7 (MDFA-Inclusion).
Input: Two MDFAs $M_1 = \langle Q_1, \Sigma, \delta_1, I_1, F_1\rangle$ and $M_2 = \langle Q_2, \Sigma, \delta_2, I_2, F_2\rangle$.
Question: Does $\mathcal{L}(M_1) \subseteq \mathcal{L}(M_2)$?

Theorem 4.8. MDFA-Inclusion *is* PSpace-complete .

Proof. As before, since MDFAs are a special case of NFAs, inclusion can be decided in PSpace [25].

For hardness, we provide a polynomial-time reduction from the MDFA - Universality problem, which is PSpace-complete by Theorem 4.6. Given an MDFA M over Σ, we construct an instance $\langle U, M\rangle$ of MDFA-Inclusion, where U is a fixed MDFA such that $\mathcal{L}(U) = \Sigma^\star$. Clearly,

$$\mathcal{L}(U) \subseteq \mathcal{L}(M) \quad \text{iff} \quad \mathcal{L}(M) = \Sigma^\star.$$

The construction of U can be performed in constant time. Hence, MDFA-Inclusion is PSpace-hard, thus finishing the proof. $\square$

5 Conclusion

In this paper, we have investigated the computational and operational state complexity of MDFAs, a relatively underexplored model of computation characterized by a highly restricted form of nondeterminism.

Overall, this work contributes to a deeper understanding of limited nondeterminism and of the structural properties of MDFAs. On one hand, the restricted nondeterminism available to MDFAs suffices to improve the state complexity bounds for union, suffix closure, and factor closure over the deterministic case, while leaving the bounds for intersection, reversal, and prefix closure unchanged. On the other hand, even this minimal degree of nondeterminism suffices to render complementation as hard as in the general NFA case. Moreover, we proved that initial state nondeterminism keeps emptiness easy, while rendering universality and inclusion intractable. Table 1 and Table 2 summarize our results.

Table 1: Summary of operational state complexity results.

Operation	DFA	$\|\Sigma\|$	MDFA	$\|\Sigma\|$	NFA	$\|\Sigma\|$
$L \cap K$	nm	2	nm	2	nm	2
$L \cup K$	nm	2	$n+m$	2	$n+m$	2
$\overline{L}$	n	1	2^n	4	2^n	2
$\mathrm{Suff}(L)$	2^n-1	2	n	2	n	2
$\mathrm{Pref}(L)$	n	2	n	2	n	2
$\mathrm{Fact}(L)$	2^{n-1}	2	n	2	n	2
L^R	2^n-1	2	2^n-1	$\mathcal{O}(n^2)$	n	2

Table 2: Summary of the complexity of decision problems.

Problem	DFA	MDFA	NFA
Membership	L	L	NL-complete
Emptiness	NL-complete	NL-complete	NL-complete
Universality	NL-complete	PSpace-complete	PSpace-complete
Inclusion	P	PSpace-complete	PSpace-complete
Minimization	P	PSpace-complete	PSpace-complete

Several questions remain open. Most notably, tight bounds for the concatenation, Kleene star, and Kleene plus operations have yet to be established, and we expect these to require novel lower bound methods. More broadly, a complete characterization of the operations for which MDFAs offer a complexity advantage over DFAs, and those for which they do not, remains an interesting direction for future work.

References

[1] Jean-Camille Birget. "Intersection and union of regular languages and state complexity". In: *Information Processing Letters* 43.4 (1992), pp. 185–190. DOI: 10.1016/0020-0190(92)90198-5.

[2] Jean-Camille Birget. "Partial orders on words, minimal elements of regular languages, and state complexity". In: *Theoretical Computer Science* 119.2 (1993), pp. 267–291. DOI: 10.1016/0304-3975(93)90160-U.

[3] Angelo Borsotti et al. "Minimizing speculation overhead in a parallel recognizer for regular texts". In: *Symposium on Principles and Practice of Parallel Programming*. 2024. DOI: 10.1145/3710848.3710866.

[4] Angelo Borsotti et al. "Multi-entry DFA with Reduced Initial States to Speedup Parallel Recognition". In: *Implementation and Application of Automata*. 2026, pp. 55–72. DOI: 10.1007/978-3-032-02602-6_5.

[5] Janusz Brzozowski. "In Search of Most Complex Regular Languages". In: *Implementation and Application of Automata*. 2012, pp. 5–24. DOI: 10.1007/978-3-642-31606-7_2.

[6] Janusz Brzozowski, Galina Jirásková, and Chenglong Zou. "Quotient Complexity of Closed Languages". In: *Theory of Computing Systems* 54.2 (2014), pp. 277–292. DOI: 10.1007/s00224-013-9515-7.

[7] János Dénes. "On transformations, transformation-semigroups and graphs". In: *Theory of Graphs: Proceedings of the Colloquium on Graph Theory held at Tihany*. 1966, pp. 65–75.

[8] Arthur Gill and Lawrence Kou. "Multiple-entry finite automata". In: *Journal of Computer and System Sciences* 9.1 (1974), pp. 1–19. DOI: 10.1016/S0022-0000(74)80034-6.

[9] Ian Glaister and Jeffrey Shallit. "A lower bound technique for the size of nondeterministic finite automata". In: *Information Processing Letters* 59.2 (1996), pp. 75–77. DOI: 10.1016/0020-0190(96)00095-6.

[10] Markus Holzer and Martin Kutrib. "Descriptional and computational complexity of finite automata—A survey". In: *Information and Computation* 209.3 (2011), pp. 456–470. DOI: 10.1016/j.ic.2010.11.013.

[11] Markus Holzer and Martin Kutrib. "State Complexity of Basic Operations on Nondeterministic Finite Automata". In: *Implementation and Application of Automata*. 2003, pp. 148–157. DOI: 10.1007/3-540-44977-9_14.

[12] Markus Holzer, Kai Salomaa, and Sheng Yu. "On the State Complexity of k-Entry Deterministic Finite Automata". In: *Journal of Automata, Languages and Combinatorics* 6.4 (2001), pp. 453–466. DOI: 10.25596/jalc-2001-453.

[13] John Hopcroft and Jeffrey Ullman. *Introduction to Automata Theory, Languages, and Computation*. 1st ed. Reading, Massachusetts: Addison-Wesley, 1979.

[14] Michal Hospodár, Galina Jirásková, and Peter Mlynárčik. "A Survey on Fooling Sets as Effective Tools for Lower Bounds on Nondeterministic Complexity". In: *Adventures Between Lower Bounds and Higher Altitudes: Essays Dedicated to Juraj Hromkovič on the Occasion of His 60th Birthday*. 2018, pp. 17–32. DOI: 10.1007/978-3-319-98355-4_2.

[15] Neil Immerman. "Nondeterministic Space Is Closed under Complementation". In: *SIAM Journal on Computing* 17.5 (1988), pp. 935–938. DOI: 10.1137/0217058.

[16] Galina Jirásková. "State complexity of some operations on binary regular languages". In: *Theoretical Computer Science* 330.2 (2005), pp. 287–298. DOI: 10.1016/j.tcs.2004.04.011.

[17] Neil D. Jones. "Space-bounded reducibility among combinatorial problems". In: *Journal of Computer and System Sciences* 11.1 (1975), pp. 68–85. DOI: 10.1016/S0022-0000(75)80050-X.

[18] Jui-Yi Kao, Narad Rampersad, and Jeffrey Shallit. "On NFAs where all states are final, initial, or both". In: *Theoretical Computer Science* 410.47 (2009), pp. 5010–5021. DOI: 10.1016/j.tcs.2009.07.049.

[19] Martin Kappes. "Descriptional Complexity of Deterministic Finite Automata with Multiple Initial States". In: *Journal of Automata, Languages and Combinatorics* 5.3 (2000), pp. 269–278. DOI: 10.25596/jalc-2000-269.

[20] Dexter Kozen. "Lower bounds for natural proof systems". In: *Symposium on Foundations of Computer Science*. 1977, pp. 254–266. DOI: 10.1109/SFCS.1977.16.

[21] Stan Lee and Steve Ditko. *Amazing Fantasy #15*. Aug. 1962.

[22] Hing Leung. "Descriptional Complexity of NFA of Different Ambiguity". In: *International Journal of Foundations of Computer Science* 16.05 (2005), pp. 975–984. DOI: 10.1142/S0129054105003418.

[23] Hing Leung. "Structurally Unambiguous Finite Automata". In: *Implementation and Application of Automata*. Ed. by Oscar H. Ibarra and Hsu-Chun Yen. 2006, pp. 198–207. DOI: 10.1007/11812128_19.

[24] Andreas Malcher. "Minimizing finite automata is computationally hard". In: *Theoretical Computer Science* 327.3 (2004), pp. 375–390. DOI: 10.1016/j.tcs.2004.03.070.

[25] Albert Meyer and Larry Stockmeyer. "The equivalence problem for regular expressions with squaring requires exponential space". In: *Symposium on Switching and Automata Theory*. 1972, pp. 125–129. DOI: 10.1109/SWAT.1972.29.

[26] Boris Mirkin. "On dual automata". In: *Cybernetics* 2 (1966), pp. 6–9. DOI: 10.1007/BF01072247.

[27] Nelma Moreira and Rogério Reis. *FAdo: Tools for Formal Languages Manipulation*. Version 2.1.3. Dec. 20, 2025. URL: https://fado.dcc.fc.up.pt/.

[28] Sophie Piccard. "Sur les fonctions définies dans les ensembles finis quelconques". In: *Fundamenta Mathematicae* 24.1 (1935), pp. 298–301. DOI: 10.4064/fm-24-1-298-301.

[29] Jean-Éric Pin. "Syntactic Semigroups". In: *Handbook of Formal Languages*. 1997, pp. 679–746. DOI: 10.1007/978-3-642-59136-5_10.

[30] Arto Salomaa, Derick Wood, and Sheng Yu. "On the state complexity of reversals of regular languages". In: *Theoretical Computer Science* 320.2 (2004), pp. 315–329. DOI: 10.1016/j.tcs.2004.02.032.

[31] Róbert Szelepcsényi. "The Method of Forced Enumeration for On-Line Recognition of Languages". In: *Acta Informatica* 26 (1988), pp. 279–284. DOI: 10.1007/BF00299636.

[32] Sheng Yu. "Regular Languages". In: *Handbook of Formal Languages*. 1997, pp. 41–110. DOI: 10.1007/978-3-642-59136-5_2.

[33] Sheng Yu, Qingyu Zhuang, and Kai Salomaa. "The state complexities of some basic operations on regular languages". In: *Theoretical Computer Science* 125.2 (1994), pp. 315–328. DOI: 10.1016/0304-3975(92)00011-F.