Weakly and Strongly Irreversible Regular Languages

Giovanna J. Lavado   Giovanni Pighizzini

Luca Prigioniero

Dipartimento di Informatica, Università degli Studi di Milano
via Comelico 39/41, 20135 Milano, Italy

{lavado, pighizzini, prigioniero}@di.unimi.it

Finite automata whose computations can be reversed, at any point, by knowing the last k symbols read from the input, for a fixed k, are considered. These devices and their accepted languages are called k-reversible automata and k-reversible languages, respectively. The existence of k-reversible languages which are not $(k-1)$-reversible is known, for each $k > 1$. This gives an infinite hierarchy of *weakly irreversible languages*, i.e., languages which are k-reversible for some k. Conditions characterizing the class of k-reversible languages, for each fixed k, and the class of weakly irreversible languages are obtained. From these conditions, a procedure that given a finite automaton decides if the accepted language is weakly or strongly (i.e., not weakly) irreversible is described. Furthermore, a construction which allows to transform any finite automaton which is not k-reversible, but which accepts a k-reversible language, into an equivalent k-reversible finite automaton, is presented.

1   Introduction

The principle of reversibility, which is fundamental in thermodynamics, has been widely investigated for computational devices. The first works on this topic already appeared half a century ago and are due to Landauer and Bennet [9, 2]. More recently, several papers presenting investigations on reversibility in space bounded Turing machines, finite automata, and other devices appeared in the literature (see, e.g., [1, 15, 6, 10, 13, 3, 11]).

A process is said to be reversible if its reversal causes no changes in the original state of the system. In a similar way, a computational device is said to be reversible when each configuration has at most one predecessor and one successor, thus implying that there is no loss of information during the computation. As observed by Landauer, logical irreversibility is associated with physical irreversibility and implies a certain amount of heat generation. Hence, in order to avoid power dissipation and to reduce the overall power consumption of computational devices, it can be interesting to realize reversible devices.

In this paper we focus on finite automata. While each two-way finite automaton can be converted into an equivalent one which is reversible [6], in the case of one-way finite automata (that, from now on, will be simply called *finite automata*) this is not always possible, namely there are regular languages as, for instance, the language a^*b^*, that are recognized only by finite automata that are not reversible [15].

In [3], the authors gave an automata characterization of the class of *reversible languages*, i.e., the class of regular languages which are accepted by reversible automata: a language is reversible if and only if the minimum deterministic automaton accepting it does not contain a certain *forbidden pattern*. Furthermore, they provide a construction to transform a deterministic automaton not containing such forbidden pattern into an equivalent reversible automaton. This construction is based on the replication of some strongly connected components in the transition graph of the minimum automaton. Unfortunately, this can lead to an exponential increase in the number of the states, which, in the worst case, cannot be avoided. To overcome this problem, two techniques for representing reversible automata, without explicitly describing replicated parts, have been obtained in [12].

In this paper, we deepen these investigations, by introducing the notions of *weakly* and *strongly irreversible* language. By definition, a reversible automaton during a computation is able to move back from a configuration (state and input head position) to the previous one by knowing the last symbol which has been read from the input tape. This is equivalent to saying that all transitions entering in a same state are on different input symbols. Now, suppose to give the possibility to automata to see back more than one symbol on the input tape, in order to move from a configuration to the previous one. Does this possibility enlarge the class of languages accepted by reversible (in this extended sense) automata? It is not difficult to give a positive answer to this question.

Considering this idea, we recall the notion of *k-reversibility*: a regular language is *k-reversible* if it is accepted by a finite automaton whose computations can be reversed by knowing the sequence of the last k symbols that have been read from the input tape. This notion was previously introduced in [8] by proving the existence of an infinite hierarchy of *degrees of irreversibility*: for each $k > 1$ there exists a language which is *k-reversible* but not $(k-1)$-reversible. Here we prove that there are regular languages which are not *k-reversible* for any k. Such languages are called *strongly irreversible*, in contrast with the other regular languages which are called *weakly irreversible*.

As in the case of "standard" reversibility (or 1-reversibility), we provide an automata characterization of the classes of weakly and strongly irreversible languages. Indeed, generalizing the notion of forbidden pattern presented in [3], we show that a language is *k-reversible* if and only if the minimum automaton accepting it does not contain a certain *k-forbidden pattern*. We also give a construction to transform each automaton which does not contain the *k-forbidden pattern*, into an equivalent automaton which is *k-reversible*. Furthermore, using a pumping argument, we prove that if an *n*-state automaton contains an *N-forbidden pattern*, for a constant $N = O(n^2)$, then it contains a *k-forbidden pattern* for each $k > 0$. Hence, applying this condition to the minimum automaton accepting a language *L*, we are able to decide if *L* is weakly or strongly irreversible. We finally present a decision procedure for such problem.

We point out that, according to the approach in [3], in this paper we refer to the *classical model* of deterministic automata, namely automata with a unique initial state, a set of final states, and deterministic transitions. Different approaches have been considered in the literature. The notion of reversibility in [1] is introduced by considering deterministic devices with one initial state and one final state, while automata with a set of initial states, a set of final states and deterministic transitions have been considered in [15]. In particular, the notion of reversibility in [1] is more restrictive than the one studied in [3] and in this paper. Hence, also the notion of *k-reversibility*, introduced and studied here, is different from a notion of *k-reversibility* studied in [1].

2   Preliminaries

In this section we recall some basic definitions and results useful in the paper. For a detailed exposition, we refer the reader to [4].

Given a set S, let us denote by $\#S$ its cardinality, by 2^S the family of all its subsets, and by $S^{<k}$ (S^k, respectively), for a fixed integer $k \geq 0$, the set of sequences of less than (exactly, resp.) k elements from S, where ε is the empty sequence. Given an alphabet Σ, $|w|$ denotes the length of a string $w \in \Sigma^*$.

A *deterministic finite automaton* (DFA) is a tuple $A = (Q, \Sigma, \delta, q_I, F)$, where Q is the finite set of *states*, Σ is the *input alphabet*, $q_I \in Q$ is the *initial state*, $F \subseteq Q$ is the set of *accepting* (or *final*) *states*, and $\delta : Q \times \Sigma \to Q$ is the partial *transition function*. A *nondeterministic finite automaton* (NFA) is an automaton in which it is possible to reach multiple states at the same time: multiple initial states are allowed and the transition function is defined as $\delta : Q \times \Sigma \to 2^Q$. The *language accepted* by an automaton

is defined in classical way as the set of all strings that define a path from one initial state to one of the accepting states.

Let $A = (Q, \Sigma, \delta, q_I, F)$ be a DFA. A state $p \in Q$ is *useful* if it is *reachable*, i.e., there exists $w \in \Sigma^*$ such that $\delta(q_I, w) = p$, and *productive*, i.e., if there is $w \in \Sigma^*$ such that $\delta(p, w) \in F$. In this paper we only consider automata with all useful states.

The *reverse* transition function of A is $\delta^R : Q \times \Sigma \to 2^Q$, with $\delta^R(p, a) = \{q \in Q \mid \delta(q, a) = p\}$. The *reverse automaton* $A^R = (Q, \Sigma, \delta^R, F, \{q_I\})$ is the NFA obtained by reversing the transition function δ and in which the set of initial states coincides with the set of final states of A and the unique final state is q_I.

A state $r \in Q$ is said to be *irreversible* when $\#\delta^R(r, a) > 1$ for some $a \in \Sigma$, i.e., there are at least two transitions on the same letter entering r, otherwise r is said to be *reversible*. The DFA A is said to be *irreversible* if it contains at least one irreversible state, otherwise A is *reversible* (REV-DFA). As pointed out in [7], the notion of reversibility for a language is related to the computational model under consideration. In this paper we only consider DFAs. Hence, by saying that a language L is *reversible*, we refer to this model, namely we mean that there exists a REV-DFA accepting L. The class of reversible languages is denoted by REV.

We say that two states $p, q \in Q$ are *equivalent* if and only if for all $w \in \Sigma^*$, $\delta(p, w) \in F$ exactly when $\delta(q, w) \in F$. Two automata A and A' are said to be *equivalent* if they accept the same language, i.e., $L(A) = L(A')$.

A *strongly connected component* (SCC) C of a NFA or a DFA A is a maximal subset of Q such that in the transition graph of A there exists a path between every pair of states in C. Let us denote by $\mathscr{C}_q$ the SCC containing the state $q \in Q$.

We consider a partial order $\preceq$ on the set of SCCs of A, such that, for two such components C_1 and C_2, $C_1 \preceq C_2$ when either $C_1 = C_2$ or no state in C_1 can be reached from a state in C_2, but a state in C_2 is reachable from a state in C_1. We write $C_1 \prec C_2$ when $C_1 \preceq C_2$ and $C_1 \neq C_2$.

3   Strong and weak irreversibility

In this section we introduce the main notions we consider in this paper, by defining strong and weak irreversibility and by presenting their basic properties.

Definition 1. *Let $k > 0$ be an integer, $A = (Q, \Sigma, \delta, q_I, F)$ be a DFA, and $L \subseteq \Sigma^*$ be a regular language.*

- *A state $r \in Q$ is said to be k-irreversible if there exist a string $x \in \Sigma^{k-1}$ and a symbol $a \in \Sigma$, such that the cardinality of the following set is greater than 1:*

$$\{\delta(p, x) \mid p \in Q \text{ and } \delta(p, xa) = r\}.$$

Otherwise, r is said to be k-reversible.

- *The automaton A is k-reversible if each of its states is k-reversible.*

- *The language L is k-reversible if there exists a k-reversible DFA accepting it.*

- *The language L is weakly irreversible if it is k-reversible for some integer $k > 0$.*

- *The language L is strongly irreversible if it is not weakly irreversible.*

By definition, a state r is 1-reversible if and only if it is reversible. As a consequence, 1-reversibility coincides with reversibility.

In the case of a k-reversible state r, with $k > 1$, we could have more than one transition on the same symbol a entering r. However, by knowing the suffix of length k of the part of the input already inspected,
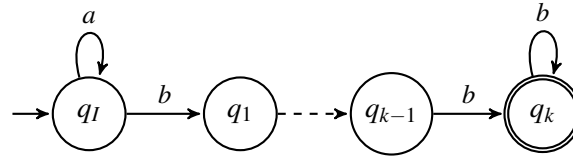
Figure 1: The minimum automaton accepting the language $a^*b^kb^*$

i.e., a suffix xa with $|x| = k - 1$, we can uniquely identify which transition on a has been used to enter r in the current computation. In other terms, while a reversible automaton is a device which is able to move the computation one state back, by knowing the last symbol that has been read, a k-reversible automaton can do the same, having the suffix of length k of the part of the input already inspected (when the length of that part is less than k, the automaton can see all the input so far inspected).

Let us denote by REV_k the class of k-reversible languages. Hence, $\mathrm{REV} = \mathrm{REV}_1$. Furthermore k-reversible DFAs are indicated as REV_k-DFAs, for short.

From Definition 1, we can immediately prove the following facts:

Lemma 1. *Let $k > 0$ be an integer, $A = (Q, \Sigma, \delta, q_I, F)$ be a DFA, and $L \subseteq \Sigma^*$ be a regular language.*

- *If a state $q \in Q$ is k-reversible, then it is k'-reversible for each $k' > k$.*

- *If a state $q \in Q$ is k-irreversible, then it is k'-irreversible for each $k' < k$.*

- *If A is k-reversible, then it is k'-reversible for each $k' > k$.*

- *If L is k-reversible, then it is k'-reversible for each $k' > k$.*

Example 1. [8] For each integer $k > 0$, consider the language $a^*b^kb^*$, which is accepted by the minimum automaton depicted in Figure 1. The only irreversible state is q_k.

Suppose that, after reading a string w, the automaton is in q_k. If we know a suffix of w of length i, with $i \leq k$, (this suffix can only be b^i) then we cannot determine the previous state in the computation, i.e., the state before reading the last symbol of w. In fact, this state could be either q_{k-1} or q_k. Hence, the automaton is not k-reversible. However, if we know the suffix of length $k + 1$, then it could be either b^{k+1}, and in this case the previous state is q_k, or ab^k, and in this case the previous state is q_{k-1}. It could be also possible that only k input symbols have been read, i.e., $|w| = k$. In that case, all $w = b^k$ can be seen back and the previous state is q_{k-1}. Hence, the automaton is $(k+1)$-reversible. As shown in [8, Theorem 4] we cannot do better for this language, i.e., $L \in \mathrm{REV}_{k+1} \setminus \mathrm{REV}_k$. This can be also obtained as a consequence of results in Section 5.

As a consequence of the last item in Lemma 1 and of Example 1 we have the proper infinite hierarchy of classes

$$\mathrm{REV} = \mathrm{REV}_1 \subset \mathrm{REV}_2 \subset \cdots \subset \mathrm{REV}_k \subset \cdots$$

In [3], the authors proved that a regular language is irreversible if and only if the minimum DFA accepting it contains a *forbidden pattern*, which consists of two transitions of a same letter entering in a same state r, where one of them arrives from a state p which belongs to the same strongly connected component of r. We now refine such definition in order to consider strings of the same length that lead to the same state:

Definition 2. *Given a DFA $A = (Q, \Sigma, \delta, q_I, F)$ and an integer $k > 0$, the k-forbidden pattern is formed by three states $p, q, r \in Q$, with a symbol $a \in \Sigma$, two strings $x \in \Sigma^{k-1}$ and $w \in \Sigma^*$, such that $p \neq q$, $\delta(p, x) \neq \delta(q, x)$, $\delta(p, xa) = \delta(q, xa) = r$, and $\delta(r, w) = q$.*
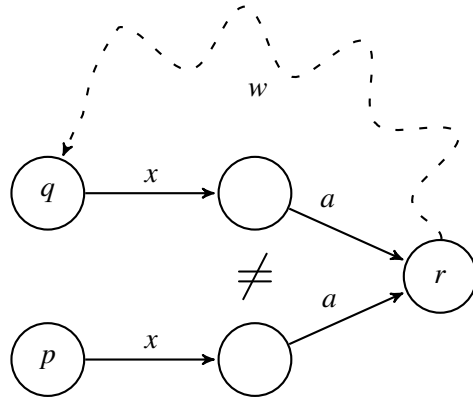
Figure 2: The k-forbidden pattern: $x \in \Sigma^{k-1}$, $a \in \Sigma$, $w \in \Sigma^*$

The k-forbidden pattern just defined is depicted in Figure 2.

From Definition 2, we can observe that if a DFA A contains a k-forbidden pattern, for some $k > 0$, then it contains a k'-forbidden pattern for each k', with $0 < k' < k$.

The notion of k-forbidden pattern will be used in the subsequent sections to obtain a characterization of the class REV_k. In fact, we will prove that a regular language is k-reversible if and only if the minimum DFA accepting it does not contain the k-forbidden pattern.

4 k-reversible simulation

In this section we present a construction to build, given a DFA $A = (Q, \Sigma, \delta, q_I, F)$ and an integer $k > 0$, an equivalent DFA $A' = (Q', \Sigma, \delta', q'_I, F')$, which is k-reversible if A does not contain the k-forbidden pattern. The DFA A' simulates A by storing in its finite control three elements:

- The current state q of A.

- An integer $j \in \{1, \ldots, k\}$ which is used to count the first k visits to states in the current SCC of A, namely in the SCC which contains the current state q. When the value of the counter reaches k, it is no more incremented, until a transition leaving the SCC. At that point, after saving its value in the third component of the state, 1 is assigned to the counter for denoting the first visit in the SCC just reached.

- A sequence of pairs from $Q \times \{1, \ldots, k\}$. This is the sequence of the first two components of the states in Q' which have been reached before simulating a transition that in A changes SCC. Since the number of possible SCCs is bounded by $\#Q$, we consider sequences of length less than $\#Q$.

Formally, we give the following definition:

- $Q' = Q \times \{1, \ldots, k\} \times (Q \times \{1, \ldots, k\})^{< \#Q}$,

- for $\langle q, j, \alpha \rangle \in Q'$, if $\delta(q, a) = p$ then

$$\delta'(\langle q, j, \alpha \rangle, a) = \begin{cases} \langle p, \min\{j+1, k\}, \alpha \rangle & \text{if } \mathscr{C}_p = \mathscr{C}_q \\ \langle p, 1, \alpha \cdot (q, j) \rangle & \text{otherwise,} \end{cases}$$

while $\delta'(\langle q, j, \alpha \rangle, a)$ is not defined when $\delta(q, a)$ is not defined, and $\cdot$ denotes the concatenation of a pair at the end of sequence,

- $q_I' = \langle q_I, 1, \varepsilon \rangle$ is the initial state,
- $F' = F \times \{1, \ldots, k\} \times (Q \times \{1, \ldots, k\})^{<\#Q}$ is the set of final states.

Notice that by dropping the second and the third components off the states of A', we get exactly the automaton A. Hence, A and A' are equivalent.

Furthermore, observe that if $\delta'(\langle p, h, \alpha \rangle, a) = \langle r, \ell, \gamma \rangle$, with $\langle p, h, \alpha \rangle, \langle r, \ell, \gamma \rangle \in Q'$, $a \in \Sigma$, and $1 < \ell \leq k$, then the states p and r are in the same SCC of A and $h = \ell - 1$ when $\ell < k$, while $h \in \{k - 1, k\}$ when $\ell = k$. This fact will be used in the following proof of the main property of A'.

Lemma 2. *If A does not contain the k-forbidden pattern, then A' is a k-reversible* DFA.

Proof. By contradiction, let us suppose that A' contains a k-irreversible state $\langle r, \ell, \gamma \rangle \in Q'$. Then there exist a string $x \in \Sigma^{k-1}$, a symbol $a \in \Sigma$, states $\langle p_0, h_0, \alpha_0 \rangle, \langle p, h, \alpha \rangle, \langle q_0, j_0, \beta_0 \rangle, \langle q, j, \beta \rangle \in Q'$, such that $\delta'(\langle p_0, h_0, \alpha_0 \rangle, x) = \langle p, h, \alpha \rangle$, $\delta'(\langle q_0, j_0, \beta_0 \rangle, x) = \langle q, j, \beta \rangle$, $\delta'(\langle p, h, \alpha \rangle, a) = \delta'(\langle q, j, \beta \rangle, a) = \langle r, \ell, \gamma \rangle$, and $\langle p, h, \alpha \rangle \neq \langle q, j, \beta \rangle$. The situation is summarized in the following picture:

$$
\begin{array}{ccc}
\langle p_0, h_0, \alpha_0 \rangle \xrightarrow{\ x\ } \langle p, h, \alpha \rangle \searrow^{a} & & \\
\neq & & \langle r, \ell, \gamma \rangle \\
\langle q_0, j_0, \beta_0 \rangle \xrightarrow{\ x\ } \langle q, j, \beta \rangle \nearrow^{a} & &
\end{array}
$$

For $k > 1$, the proof is divided in three cases, depending on the value of ℓ.

- *Case $\ell = 1$.*
  Considering the definition of δ', we notice that both states p and q are not in the same SCC of r. Then $\gamma = \alpha \cdot (p, h) = \beta \cdot (q, j)$, thus implying $\alpha = \beta$, $p = q$, and $h = j$. This is a contradiction to the hypothesis $\langle p, h, \alpha \rangle \neq \langle q, j, \beta \rangle$.

- *Case $1 < \ell < k$.*
  Again from the definition of δ', we can observe that $h = j = \ell - 1 < k - 1$ and $\alpha = \beta = \gamma$. We decompose x as $x'bx''$, where $x', x'' \in \Sigma^*$, $b \in \Sigma$ and $|x''| = \ell - 2$. Then, in the paths on the string x from $\langle p_0, h_0, \alpha_0 \rangle$ to $\langle p, \ell - 1, \alpha \rangle$ and from $\langle q_0, j_0, \beta_0 \rangle$ to $\langle q, \ell - 1, \alpha \rangle$ the last transitions that change SCC in A are those on the symbol b, immediately after the prefix x', i.e, we have the following situation:

$$
\begin{array}{ccccccc}
\langle p_0, h_0, \alpha_0 \rangle \xrightarrow{\ x'\ } \langle p_1, h_1, \alpha_1 \rangle \xrightarrow{\ b\ } \langle p_2, 1, \alpha \rangle \xrightarrow{\ x''\ } \langle p, \ell - 1, \alpha \rangle \searrow^{a} & & \\
& & \langle r, \ell, \alpha \rangle \\
\langle q_0, j_0, \beta_0 \rangle \xrightarrow{\ x'\ } \langle q_1, j_1, \beta_1 \rangle \xrightarrow{\ b\ } \langle q_2, 1, \alpha \rangle \xrightarrow{\ x''\ } \langle q, \ell - 1, \alpha \rangle \nearrow^{a} & &
\end{array}
$$

  for suitable $\langle p_1, h_1, \alpha_1 \rangle, \langle q_1, j_1, \beta_1 \rangle \in Q'$, $p_2, q_2 \in Q$. Then $\alpha = \alpha_1 \cdot (p_1, h_1) = \beta_1 \cdot (q_1, j_1)$, that implies $p_1 = q_1$. As a consequence, since A is deterministic we get that $p_2 = q_2$ and $p = q$. Thus, also in this case we get the contradiction $\langle p, h, \alpha \rangle = \langle q, j, \beta \rangle$.

- *Case $\ell = k$.*
  From the definition of δ', we notice that either $h = j = k - 1$, or at least one of h and j is equal to k. In the first case, the proof can be completed as in the case $1 < \ell < k$, leading to a contradiction. In the case $h = k$, moving backwards from the state $\langle p, k, \alpha \rangle$ to $\langle p_0, h_0, \alpha_0 \rangle$, along the transitions on the string x of length $k - 1$, we find a sequence of states whose all second components are equal to k, which is followed by a (possibly empty) sequence of states where the values of the second

components decrease by 1 at each transition. In this way we can conclude that $h_0 \geq 1$ and all the first components, included p_0, of states on this path, are in the same SCC of r. Hence, A contains the k-forbidden pattern. The case $j = k$ is similar.

For $k = 1$, if p or q are in the same SCC as r then A should contain the 1-forbidden pattern. Otherwise, we can proceed as in the case $\ell = 1$, obtaining a contradiction. □

We now evaluate the size of the automaton obtained by using the previous construction.

Theorem 1. *Each n-state DFA which does not contain the k-forbidden pattern can be simulated by an equivalent k-reversible DFA with no more than $(k+1)^{n-1}$ states.*

Proof. Let A be a given n-state DFA not containing the k-forbidden pattern. According to Lemma 2, the automaton A' obtained from A with the above presented construction is k-reversible. Now, we are going to estimate the number of reachable states in it.

First of all, we notice that if $\langle q, \ell, \alpha \rangle$ is a reachable state of A' and $\alpha = ((p_1, j_1), (p_2, j_2), \ldots, (p_h, j_h))$, then $\mathscr{C}_{p_1} \prec \mathscr{C}_{p_2} \prec \cdots \prec \mathscr{C}_{p_h} \prec \mathscr{C}_q$. Hence, since the ordering among states appearing in α is given by the ordering of SCCs in A, we could represent α as a set.

This also allows to interpret the state $\langle q, \ell, \alpha \rangle$ as the function $f : Q \to \{0, 1, \ldots, k\}$, such that for $r \in Q$:

$$f(r) = \begin{cases} \ell & \text{if } r = q, \\ j_i & \text{if } r = p_i, \, 1 \leq i \leq h, \\ 0 & \text{otherwise.} \end{cases}$$

By counting the number of possible functions, we obtain a $(k+1)^n$ upper bound for the number of reachable states in A'.

Now, we show how to reduce this bound to the one claimed in the statement of the theorem.

The above presented simulation can be slightly refined by observing that while simulating states in the SCC of the initial state q_I, it is not necessary to keep the counter. Furthermore, in each state $\langle q, \ell, \alpha \rangle$ of Q', with $q \notin \mathscr{C}_{q_I}$, the first element of α, which should represent a state in $\mathscr{C}_{q_I}$, is stored without the counter. Hence, the state $\langle q, \ell, \alpha \rangle$ can be seen as a state in $\mathscr{C}_{q_I}$ (the first element of α) with a function $f : Q \setminus \mathscr{C}_{q_I} \to \{0, 1, \ldots, k\}$ (representing the current state with its counter and the other pairs in α). Since the counter associated with the current state is always positive, f cannot be the null function. Hence, the number of possible functions is bounded by $(k+1)^{n-s} - 1$, where $s = \#\mathscr{C}_{q_I}$. Considering also the states which are used in Q' to simulate the states in $\mathscr{C}_{q_I}$, this gives at most $s + s((k+1)^{n-s} - 1)$ many reachable states. For $k > 0$ this amount is bounded by $(k+1)^{n-1}$. □

We point out that for $k = 1$, Theorem 1 gives a 2^{n-1} upper bound, which matches with the bound for the conversion of DFAs into equivalent REV-DFAs, claimed in [3]. In the same paper, a lower bound very close to such an upper bound was presented.

5 A characterization of *k*-reversible languages

In this section we present a characterization of k-reversible languages based on the notion of k-forbidden pattern. This characterization will be obtained by combining Theorem 1 with the following result.

Lemma 3. *Let L be a regular language and k be a positive integer. If the minimum DFA accepting L contains the k-forbidden pattern, then $L \notin \text{REV}_k$.*
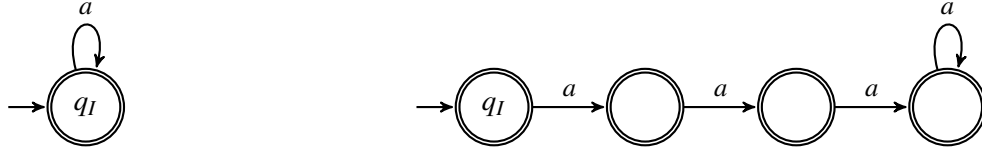
Figure 3:   The minimum DFA accepting the reversible language a^*, and an equivalent DFA containing the 3-forbidden pattern

Proof. Let $M = (Q, \Sigma, \delta, q_I, F)$ be the minimum DFA accepting L. By hypothesis there exist $p, q, r \in Q$, $a \in \Sigma$, $x \in \Sigma^{k-1}$, $w \in \Sigma^*$ such that $p \neq q$, $\delta(p, x) \neq \delta(q, x)$, $\delta(p, xa) = \delta(q, xa) = r$ and $\delta(r, w) = q$. Let $s = \delta(p, x)$ and $t = \delta(q, x)$. We are going to prove that each DFA $A' = (Q', \Sigma, \delta', q'_I, F')$ accepting L contains a k-irreversible state.

Let $q_0 \in Q'$ be a state equivalent to p. In A' we consider two arbitrarily long sequences of states $q_1, q_2, \ldots$ and $r_1, r_2, \ldots$ equivalent to q and r, respectively, such that $\delta'(q_{h-1}, xa) = r_h$ and $\delta'(r_h, w) = q_h$, for $h > 0$. Since Q' is finite, sooner or later we will find an index j such that either $r_i = r_j$ or $q_i = q_j$, for some $1 \leq i < j$. Let us take the first j with such property.

- Suppose $r_i = r_j$. If $i = 1$, let $\hat{s} = \delta'(q_0, x)$ and $\hat{t} = \delta'(q_{j-1}, x)$. Since q_0 is equivalent to p and q_{j-1} is equivalent to q, $\hat{s}$ and $\hat{t}$ are equivalent to the states s and t of M, respectively. So, $\hat{s} \neq \hat{t}$. Furthermore, $\delta'(\hat{s}, a) = \delta'(\hat{t}, a) = r_1$. Hence, r_1 is k-irreversible. In the case $i > 1$, since j is the first index giving a repetition we get $q_{i-1} \neq q_{j-1}$. We decompose the string xa as $x'\gamma x''$, where $x', x'' \in \Sigma^*$, $\gamma \in \Sigma$, and $\delta'(q_{i-1}, x') \neq \delta'(q_{j-1}, x')$, $\delta'(q_{i-1}, x'\gamma) = \delta'(q_{j-1}, x'\gamma) = u$ for some $u \in Q'$ and $\delta'(u, x'') = r_i$. We observe that $\delta'(q_{i-2}, xawx') = \delta'(q_{i-1}, x') \neq \delta'(q_{j-2}, xawx') = \delta'(q_{j-1}, x')$, while $\delta'(q_{i-2}, xawx'\gamma) = \delta'(q_{j-2}, xawx'\gamma) = u$. This implies that the state u is $|xawx'\gamma|$-irreversible. Hence it is k-irreversible.

- In the case $q_i = q_j$ and $r_i \neq r_j$, we observe that since q_0 is equivalent to p and q_j is equivalent to q for $j \geq 1$, while p and q are not equivalent, we get $i > 0$. We decompose w as $w'\gamma w''$, where $w', w'' \in \Sigma^*$, $\gamma \in \Sigma$ and $\delta'(r_i, w') \neq \delta'(r_j, w')$, $\delta'(r_i, w'\gamma) = \delta'(r_j, w'\gamma) = u$ for some $u \in Q'$ and $\delta'(u, w'') = q_i$. Then, $\delta'(q_{i-1}, xaw') \neq \delta'(q_{j-1}, xaw')$ and $\delta'(q_{i-1}, xaw'\gamma) = \delta'(q_{j-1}, xaw'\gamma) = u$. Hence, the state u is $|xaw'\gamma|$-irreversible, so it is k-irreversible. □

Notice that the condition in Lemma 3 is on the *minimum* DFA accepting the language under consideration. If we remove the requirement that the considered DFA has to be minimum, the statement becomes false. For instance, the language $L = a^*$ is reversible even though for each $k > 0$ we can build a DFA accepting it, which contains the k-forbidden pattern (see Figure 3).

We are now able to characterize k-reversible languages in terms of the structure of minimum DFAs:

Theorem 2. *Let L be a regular language. Given $k > 0$, $L \in \text{REV}_k$ if and only if the minimum DFA accepting L does not contain the k-forbidden pattern.*

Proof. The *if* part is a consequence of Theorem 1, the *only-if* part derives from Lemma 3. □

From Theorem 2, we observe that to transform each DFA A accepting a k-reversible language into an equivalent REV_k-DFA, firstly we can transform A into the equivalent minimum DFA M and then we can apply to M the construction presented in Section 4.

As a consequence of Theorem 2 we also obtain:

Corollary 1. $L \in \text{REV}_{k+1} \setminus \text{REV}_k$ *if and only if the maximum h such that the minimum* DFA *accepting L contains the h-forbidden pattern is k.*

In the following result we present further families of languages, besides that in Example 1, which witness the existence of the proper infinite hierarchy

$$\text{REV} = \text{REV}_1 \subset \text{REV}_2 \subset \cdots \subset \text{REV}_k \subset \cdots$$

Furthermore, we show that the difference between the "amount" of irreversibility in a minimum DFA and in the accepted language can be arbitrarily large:

Theorem 3. *For all integers $k, j > 0$ with $j > k > 1$ there exists a language $L_{k,j}$ such that:*

- *The minimum* DFA *accepting $L_{k,j}$ is a* REV_j-DFA *but not a* REV_{j-1}-DFA.

- *$L_{k,j} \in \text{REV}_k \setminus \text{REV}_{k-1}$.*

Proof. Let $L_{k,j}$ be the language accepted by the automaton $A_{k,j} = (Q, \Sigma, \delta, q_I, F)$ where $\Sigma = \{a, b\}$, $Q = \{q_I, q_1', q_1'', \ldots, q_{j-1}', q_{j-1}'', q_j\}$, $F = \{q_{j-1}'', q_j\}$, and the transition function is defined as follows (see Figure 4 for an example):

- $\delta(q_I, a) = q_1'$
- $\delta(q_I, b) = q_1''$
- $\delta(q_i', a) = q_{i+1}'$ and $\delta(q_i'', a) = q_{i+1}''$ for $1 \leq i \leq j - k$
- $\delta(q_i', b) = q_{i+1}'$ and $\delta(q_i'', b) = q_{i+1}''$ for $j - k < i < j - 1$
- $\delta(q_{j-1}', b) = \delta(q_{j-1}'', b) = \delta(q_{j-1}'', a) = \delta(q_j, b) = q_j$

Firstly, we can observe that $A_{k,j}$ is the minimum DFA accepting $L_{k,j}$. It contains only one irreversible state, q_j, with $\delta^R(q_j, b) = \{q_j, q_{j-1}', q_{j-1}''\}$. We also notice that $\delta(q_1', a^{j-k}b^{k-2}) = q_{j-1}' \neq q_{j-1}'' = \delta(q_1'', a^{j-k}b^{k-2})$, while $\delta(q_1', a^{j-k}b^{k-1}) = \delta(q_1'', a^{j-k}b^{k-1}) = q_j$. Hence $A_{k,j}$ is not a REV_{j-1}-DFA. However, the knowledge of one more symbol in the suffix of the input read to enter q_j allows to determine the state of the automaton before reading the last symbol. In particular, if the suffix of length j is $a^{j-k+1}b^{k-1}$, then the state was q_{j-1}'; if the suffix is $ba^{j-k}b^{k-1}$ or $ba^{j-k}b^{k-2}a$, then the state was q_{j-1}''; in the remaining cases it was q_j. Hence, $A_{k,j}$ is a REV_j-DFA.

To prove that $L_{k,j} \in \text{REV}_k \setminus \text{REV}_{k-1}$, we first show that $A_{k,j}$ contains the $(k-1)$-forbidden pattern. To this aim, in Definition 2 we can choose $q = r = q_j$, $p = q_{j-k+1}'$, $a = b$, $x = b^{k-2}$ and $w = \varepsilon$. Furthermore, it is possible to obtain a REV_k-DFA $A_{k,j}'$ equivalent to $A_{k,j}$ by duplicating q_j with its loop and by redistributing incoming transitions from q_{j-1}' and q_{j-1}'', as in the case presented in Figure 5. $\square$

6 Weakly and strongly irreversible languages

By Definition 1, a language is *weakly irreversible* if it is k-reversible for some $k > 0$, namely if it is in the class $\bigcup_{k>0} \text{REV}_k$. A natural question is whether or not the class of weakly irreversible languages coincides with the class of regular languages. In this section we will give a negative answer to this question, thus proving the existence of strongly irreversible languages.

First of all, we observe that, by Theorem 2, a regular language is *strongly irreversible* if and only if the minimum DFA accepting it contains a k-forbidden pattern for each $k > 0$. Using a combinatorial argument, we now prove that in order to decide if a language is strongly or weakly irreversible, it is enough to consider only a value of k which depends on the size of the minimum DFA:
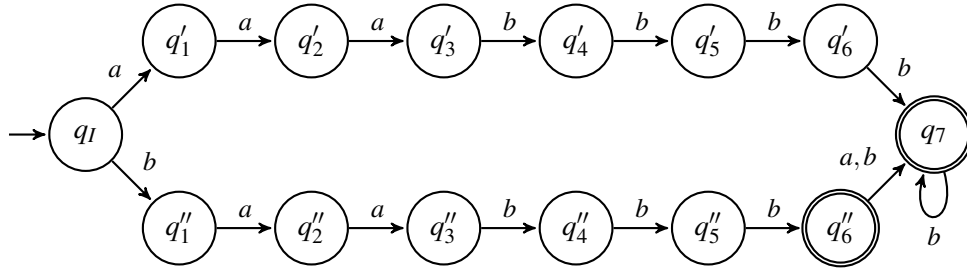
Figure 4: The minimum automaton $A_{5,7}$ accepting the language $L_{5,7}$
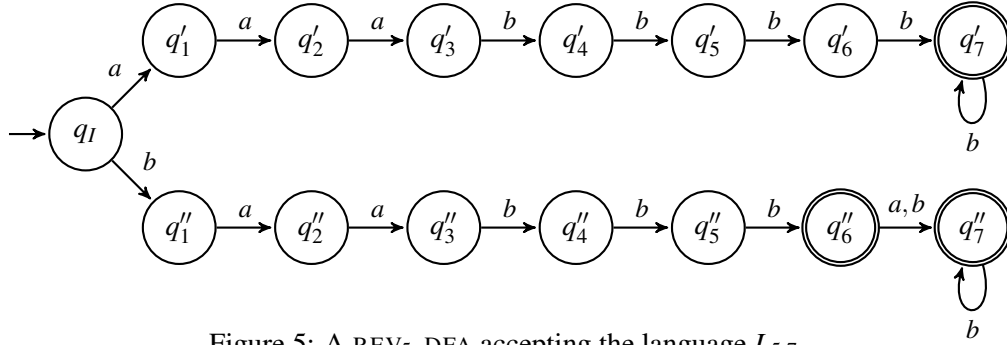
Figure 5: A REV$_5$-DFA accepting the language $L_{5,7}$

Theorem 4. *Let $A = (Q, \Sigma, \delta, q_I, F)$ be an n-state DFA and $N > \frac{n^2-n}{2}$. If A contains an N-forbidden pattern, then it contains a k-forbidden pattern for each $k > 0$.*

Proof. Suppose that A contains an N-forbidden pattern. As observed after Definition 2, A contains a k-forbidden pattern for each $k \leq N$.

We now prove the same for $k > N$. By hypothesis there exist $p, q, r \in Q$, $a \in \Sigma$, $x \in \Sigma^{N-1}$, $w \in \Sigma^*$, such that $p \neq q$, $\delta(p, x) \neq \delta(q, x)$, $\delta(p, xa) = \delta(q, xa) = r$, and $\delta(r, w) = q$. Let $x = a_1 a_2 \cdots a_{N-1}$ with $a_i \in \Sigma$, for $i = 1, \ldots, N-1$. Moreover, let $p_0, \ldots, p_{N-1}, q_0, \ldots, q_{N-1} \in Q$ be such that $p = p_0$, $q = q_0$, $p_i = \delta(p_{i-1}, a_i)$, $q_i = \delta(q_{i-1}, a_i)$ for $i = 1, \ldots, N-1$, and $\delta(p_{N-1}, a) = \delta(q_{N-1}, a) = r$. Since $p_{N-1} \neq q_{N-1}$ and A is deterministic, we get $p_i \neq q_i$ for $i = 0, \ldots, N-1$. Notice that there are $n^2 - n$ possible pairs of different states.

We consider the pairs $(p_0, q_0), \ldots, (p_{N-1}, q_{N-1})$. Since $N > (n^2 - n)/2$ and $p_i \neq q_i$, for $i = 0, \ldots, N-1$, there are two indices i, j, $0 \leq i < j \leq N-1$ such that either $(p_i, q_i) = (p_j, q_j)$ or $(p_i, q_i) = (q_j, p_j)$. So $\delta(p_i, (a_{i+1} \cdots a_j)^2) = p_i$ and $\delta(q_i, (a_{i+1} \cdots a_j)^2) = q_i$. Given $h > 0$, we consider the string $z_h = a_1 \cdots a_i (a_{i+1} \cdots a_j)^{2h} a_{j+1} \cdots a_{N-1}$. We can verify that $\delta(p, z_h) = p_{N-1}$ and $\delta(q, z_h) = q_{N-1}$. This implies that A contains the $|z_h| + 1$-forbidden pattern. Since $i \neq j$, by properly choosing h, this allows us to obtain a k-forbidden pattern for each arbitrarily large k. $\qquad \square$

Combining Theorem 2 with Theorem 4 we obtain:

Corollary 2. *Let L be a regular language whose minimum DFA has n states. Then L is strongly irreversible if and only if it is not $(\frac{n^2-n}{2} + 1)$-reversible.*

We now present an example of strongly irreversible language.

Example 2. The language $L = a^*b(a+b)^*$ is strongly irreversible. The minimum automaton accepting it has 2 states (see Figure 6). We notice that $\delta(q_I, ab) = \delta(p, ab) = p$, while $\delta(q_I, a) \neq \delta(p, a)$. This defines
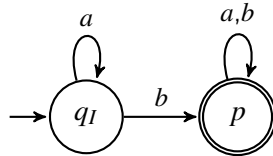
Figure 6: The minimum automaton accepting the language $L = a^*b(a+b)^*$

a 2-forbidden pattern. According to Corollary 2, this implies that L is strongly irreversible. Observe that entering in p with each string $a^k b$, we have a $(k+1)$-forbidden pattern, for any $k \geq 0$.

7   Decision problems

In this section we provide a method to decide whether a language L is strongly or weakly irreversible, and, in the latter case, to find the minimum k such that L is k-reversible.

   The idea is to simultaneously analyze all the paths entering each irreversible state $r \in Q$ of the minimum automaton A accepting L in order to find the longest string z that, with at least two different paths, leads to r and defines the $|z|$-forbidden pattern or to discover that there exist arbitrarily long strings with such property. This corresponds to analyze all couples of paths starting from two different states $p, q \in Q$ that, with the same string z, lead to r. Intuitively, this can be done by constructing the product automaton of two copies of the reversal automaton of A, i.e., $A^R \times A^R$, and by analyzing all paths starting from the states of the form (r, r). Since the goal is to establish the nature of the (ir)reversability of L — not of A — it is useful to recall that by Definition 2 it is enough to consider only the couples of paths in which one of them is completely included in the same SCC of r, i.e., $\mathscr{C}_r = \mathscr{C}_q$. To this aim, we are going to consider the product between A^R and a transformation of A^R which is obtained by splitting it in SCCs.

   Let $A = (Q, \Sigma, \delta, q_I, F)$ be an irreversible DFA, $A^R = (Q, \Sigma, \delta^R, F, \{q_I\})$ be the reversal automaton of A, and $A^R_{\text{SCCs}} = (Q, \Sigma, \delta^R_{\text{SCCs}}, F, \{q_I\})$ be the NFA obtained by splitting A^R in its SCCs, i.e., $\delta^R_{\text{SCCs}}(r, a) = \{q \mid q \in \delta^R(r, a) \text{ and } \mathscr{C}_r = \mathscr{C}_q\}$, for $r \in Q$, $a \in \Sigma$. Let us define the automaton $\hat{A} = A^R \times A^R_{\text{SCCs}}$ as follows: $\hat{A} = (\hat{Q}, \Sigma, \hat{\delta}, \hat{I}, \hat{F})$ where $\hat{Q} = \hat{F} = Q \times Q$, $\hat{I} = \{(r, r) \mid r \in Q\}$, and $\hat{\delta}((r', r''), a) = \{(p, q) \in \delta^R(r', a) \times \delta^R_{\text{SCCs}}(r'', a) \mid p \neq q\}$.

   The resulting automaton $\hat{A}$ accepts all strings z which define a $|z|$-forbidden pattern (plus the empty string). Formally, this follows from the following lemma, whose proof can be given by induction:

Lemma 4. *Consider a path* $(r, r), (p_1, q_1), \ldots, (p_{|z|-1}, q_{|z|-1}), (p, q)$ *in* $\hat{A}$ *from a state* (r, r) *to* (p, q) *on a string* z. *Then* $\hat{\delta}((r, r), z) \ni (p, q)$ *if and only if all the following conditions are satisfied:*

   1. $p_i \neq q_i$ *for each* $0 < i < |z|$, $p \neq q$,

   2. $\delta(p, z) = r$,

   3. $\delta(q, z) = r$ *and* $\mathscr{C}_r = \mathscr{C}_q$.

   Considering Theorem 2, this leads to state the following

Lemma 5. *Let A be a minimum n-state DFA and $\hat{A}$ be the NFA defined as above. Then:*

   • *The following statements are equivalent:*

      – *A is strongly irreversible,*

   – $L(\hat{A})$ *is an infinite language,*

   – $L(\hat{A})$ *contains a string of length* $\frac{n^2-n}{2}+1$.

- *For each* $k > 0$, $L(A) \in \mathrm{REV}_k$ *if and only if* $L(\hat{A})$ *contains only strings of length less than k.*

The same argument can be exploited to prove that the problem of checking whether $L(A)$ is strongly or weakly irreversible is in NL, namely the class of problems accepted by nondeterministic logarithmic space bounded Turing machines.

Theorem 5. *The problem of deciding whether a language is strongly or weakly irreversible is* NL-*complete.*

Proof. (sketch) Given a minimum DFA accepting the language under consideration and the above described automaton $\hat{A}$, the problem can be reduced to testing if the transition graph of $\hat{A}$ contains at least one loop. In such a case, there are arbitrarily long strings in $L(\hat{A})$, namely strings describing k-forbidden patterns for arbitrarily large k, and $L(A)$ is strongly irreversible. The problem of verifying the existence of a loop is in NL.

To prove the NL-completeness, we show a reduction from the *Graph Accessibility Problem* (*GAP*) which is NL-complete (for further details see [5]). Let $G = (V,E)$ be a directed graph where $V = \{1,\ldots,n\}$. Our goal is to define a DFA A such that A is strongly irreversible if and only if there exists a path from 1 to n in G. We build A' by starting from the same "state structure" of G, and adding a SCC providing the forbidden pattern when combined with a path from 1 to n in the original graph.

We stress that the instance of our problem should be an automaton containing only useful states, while automata that can be "intuitively" obtained from GAP instances could have useless states and, detecting them, would require to solve GAP.

Let $A' = (Q, \Sigma, \delta, q_I, \{q_F\})$ be a DFA where $Q = V \cup \{q_I, q_F, q_1, \ldots, q_{n-1}\}$, $\Sigma = \{0,\ldots,n,\$,\sharp\}$, and δ is defined as follows:

   i. $\delta(i,j) = j$ for $(i,j) \in E$, $i \neq j$

   ii. $\delta(q_i,j) = q_j$ for $0 < i, j < n$, $i \neq j$

   iii. $\delta(q_i,n) = n$ for $0 < i < n$

   iv. $\delta(n,0) = q_1$

   v. $\delta(q_I,i) = i$ and $\delta(i,\sharp) = q_F$ for $0 < i \leq n$

   vi. $\delta(1,\$) = 1$ and $\delta(q_1,\$) = q_1$.

Observe that the restriction of the underlying graph A' to states $1,\ldots,n$ coincides with G (transitions i.). In addition, the set of states $\{q_1,\ldots,q_{n-1}\}$ extends the SCC $\mathscr{C}_n$ so that each state can reach the others in $\mathscr{C}_n$ with a single transition (transitions ii., iii., and iv.). This implies that the state n is reachable from q_1 with all the possible paths passing through the states in the SCC. Furthermore, a loop is added to states 1 and q_1 on the symbol $\$$ in order to create a forbidden pattern (transitions vi.). Notice that each state in Q is useful (transitions v.).

In such a way, the states $\{1,n,q_1\}$ form a forbidden pattern with strings of arbitrary length if and only if the given graph contains a path from n to 1. Notice that any state $i \in Q \setminus \{n\}$ is, at most, 1-irreversible. So we can conclude that A' is strongly irreversible if and only if there exists a path from 1 to n in G.

It can be shown that the reduction can be computed in deterministic logarithmic space. □

8 Conclusion

We introduced and studied the notions of strong and weak irreversibility for finite automata and regular languages. In Section 5 we proved the existence of an infinite hierarchy of weakly irreversible languages, while in Section 6 we showed the existence of strongly irreversible languages, namely of regular languages that are not weakly irreversible. In both cases, the witness languages are defined over a binary alphabet, so the question arises if the same results hold in the case of a one-letter alphabet, i.e., in the case of unary languages. We now briefly discuss this point.

First of all, we remind the reader that the transition graph of a unary DFA consists of an initial path, which is followed by a loop (for a recent survey on unary automata, we address the reader to [14]). Hence, a unary DFA is reversible if and only if the initial path is of length 0, i.e., the automaton consists only of a loop (in this case the accepted language is said to be *cyclic*). We can also observe that given an integer $k > 0$, a unary language is k-reversible if and only if it is accepted by a DFA with an initial path of less than k states. Hence, for each k, the language $a^{k-1}a^*$ is k-reversible, but not $(k-1)$-reversible. This shows the existence of an infinite hierarchy of weakly irreversible languages even in the unary case. Furthermore, from the above discussion, we can observe that if a unary language is accepted by a DFA with an initial path of k states, then it is $(k+1)$-reversible. This implies that each unary regular language is weakly irreversible (see also [8, Proposition 10]). Hence, to obtain strongly irreversible languages, we need alphabets of at least two letters.

The definition of k-reversible automata and languages have been given for each integer $k > 0$. One could ask if it does make sense to consider a notion of 0-reversibility. According to the interpretation we gave to k-reversibility, a state is 0-reversible when in each computation its predecessor can be obtained by knowing the last 0 symbols which have been read from the input, i.e., without the knowledge of any previous input symbol. This means that a 0-irreversible state can have only one entering transition, or no entering transitions if it is the initial state. As a consequence, the transition graph of a 0-reversible automaton is a tree rooted in the initial state and 0-reversible languages are exactly finite languages.

Acknowledgment

References

[1] Dana Angluin (1982): *Inference of Reversible Languages.* *J. ACM* 29(3), pp. 741–765, doi:10.1145/322326.322334.

[2] Charles H. Bennett (1973): *Logical Reversibility Of Computation.* *IBM Journal of Research and Development* 17(6), pp. 525–532, doi:10.1147/rd.176.0525.

[3] Markus Holzer, Sebastian Jakobi & Martin Kutrib (2015): *Minimal Reversible Deterministic Finite Automata.* In Igor Potapov, editor: *Developments in Language Theory - 19th International Conference, DLT 2015, Liverpool, UK, July 27-30, 2015, Proceedings.*, Lecture Notes in Computer Science 9168, Springer, pp. 276–287, doi:10.1007/978-3-319-21500-6_22.

[4] John E. Hopcroft & Jeffrey D. Ullman (1979): *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley.

[5] Neil D. Jones (1975): *Space-bounded reducibility among combinatorial problems*. Journal of Computer and System Sciences 11(1), pp. 68–85, doi:10.1016/S0022-0000(75)80050-X.

[6] Attila Kondacs & John Watrous (1997): *On the Power of Quantum Finite State Automata*. In: *FOCS*, IEEE Computer Society, pp. 66–75, doi:10.1109/SFCS.1997.646094.

[7] Martin Kutrib (2015): *Reversible and Irreversible Computations of Deterministic Finite-State Devices*. In Giuseppe F. Italiano, Giovanni Pighizzini & Donald Sannella, editors: *Mathematical Foundations of Computer Science 2015 - 40th International Symposium, MFCS 2015, Milan, Italy, August 24-28, 2015, Proceedings, Part I, Lecture Notes in Computer Science* 9234, Springer, pp. 38–52, doi:10.1007/978-3-662-48057-1_3.

[8] Martin Kutrib & Thomas Worsch (2014): *Degrees of Reversibility for DFA and DPDA*. In Shigeru Yamashita & Shin-ichi Minato, editors: *Reversible Computation - 6th International Conference, RC 2014, Kyoto, Japan, July 10-11, 2014. Proceedings, Lecture Notes in Computer Science* 8507, Springer, pp. 40–53, doi:10.1007/978-3-319-08494-7_4.

[9] Rolf W. Landauer (1961): *Irreversibility and Heat Generation in the Computing Process*. IBM Journal of Research and Development 5(3), pp. 183–191, doi:10.1147/rd.53.0183.

[10] Klaus-Jörn Lange, Pierre McKenzie & Alain Tapp (2000): *Reversible Space Equals Deterministic Space*. J. Comput. Syst. Sci. 60(2), pp. 354–367, doi:10.1006/jcss.1999.1672.

[11] Giovanna J. Lavado, Giovanni Pighizzini & Luca Prigioniero (2016): *Minimal and Reduced Reversible Automata*. In Cezar Câmpeanu, Florin Manea & Jeffrey Shallit, editors: *Descriptional Complexity of Formal Systems - 18th IFIP WG 1.2 International Conference, DCFS 2016, Bucharest, Romania, July 5-8, 2016. Proceedings, Lecture Notes in Computer Science* 9777, Springer, pp. 168–179, doi:10.1007/978-3-319-41114-9_13. An extended version will appear in the Journal of Automata, Languages, and Combinatorics.

[12] Giovanna J. Lavado & Luca Prigioniero (2017): *Concise Representations of Reversible Automata*. In Giovanni Pighizzini & Cezar Câmpeanu, editors: *Descriptional Complexity of Formal Systems - 19th IFIP WG 1.02 International Conference, DCFS 2017, Milano, Italy, July 3-5, 2017, Proceedings, Lecture Notes in Computer Science* 10316, Springer, pp. 238–249, doi:10.1007/978-3-319-60252-3_19.

[13] Sylvain Lombardy (2002): *On the Construction of Reversible Automata for Reversible Languages*. In Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephan Eidenbenz & Ricardo Conejo, editors: *Automata, Languages and Programming, 29th International Colloquium, ICALP 2002, Malaga, Spain, July 8-13, 2002, Proceedings, Lecture Notes in Computer Science* 2380, Springer, pp. 170–182, doi:10.1007/3-540-45465-9_16.

[14] Giovanni Pighizzini (2015): *Investigations on Automata and Languages Over a Unary Alphabet*. Int. J. Found. Comput. Sci. 26(7), pp. 827–850, doi:10.1142/S012905411540002X.

[15] Jean-Eric Pin (1992): *On Reversible Automata*. In Imre Simon, editor: *LATIN '92, 1st Latin American Symposium on Theoretical Informatics, São Paulo, Brazil, April 6-10, 1992, Proceedings, Lecture Notes in Computer Science* 583, Springer, pp. 401–416, doi:10.1007/BFb0023844.