International Journal of Foundations of Computer Science

Weakly and Strongly Irreversible Regular Languages

Bruno Guillon
LIMOS, Université Clermont-Auvergne, France
bruno.guillon@uca.fr

Giovanna J. Lavado
Giovanni Pighizzini
Luca Prigioniero
Dipartimento di Informatica, Università degli Studi di Milano
via Celoria 18, 20133 Milano, Italy
{*lavado*, *pighizzini*, *prigioniero*}*@di.unimi.it*

Received (Day Month Year)
Accepted (Day Month Year)
Communicated by (xxxxxxxxxx)

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.

Keywords: Finite automata; Reversibility; Descriptional Complexity.

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 [2, 9]. 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, 3, 6, 10, 11, 13, 15]).

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 [9]. 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 the same state are on different input symbols. Now, suppose to give the possibility to the 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 $k > 0$ or equivalently, a certain *strong forbidden pattern*. 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].

Outline. The paper is organized as follows. In Section 2, we recall the basic definitions and we introduce the main concepts under consideration (mainly, the notions of *k-reversibility*, *weak* and *strong irreversibility*). In Section 3, we prove our main result which characterizes the class of k-reversible languages, for each positive integer k, by using the notion of *k-forbidden pattern*. To prove this characterization, in Subsection 3.1 we present a construction to transform an automaton into an equivalent k-reversible one, when this is possible. The above-mentioned characterization is then extended to weakly irreversible languages (i.e., k-reversible languages, for some k) in Section 4. In Section 5 we consider the degree of irreversibility on both automata and their accepted languages, and we show the existence of arbitrarily large (and even infinite) gaps between these degrees. In Section 6, we discuss and we present a decision procedure that allows to check whether a minimum automaton accepts a weakly or a strongly irreversible language. The same procedure also allows to compute the degree of irreversibility of a language. We prove the NL-completeness of such decision problem. Finally, we make concluding comments in Section 7.

2. Preliminaries

2.1. *Automata, words and languages*

In this section we recall some basic definitions and results useful in the paper. For more details on automata and formal languages, we refer the reader to a standard textbook as, e.g., [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 *final states*, and $\delta : Q \times \Sigma \to Q$ is the partial *transition function*, that can be extended, in the usual way, as a function $\delta : Q \times \Sigma^* \to Q$ applying to strings. 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 $\delta(q, w) \in F$. 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 DFAs in which all the states are useful.

Nondeterministic finite automata (NFAs) are defined by extending the transition function to $\delta : Q \times \Sigma \to 2^Q$ and by admitting several initial states (the initial state component is replaced by a subset $I \subseteq Q$ of initial states). In this way, an NFA can reach multiple states at the same time.

The *language accepted by a* DFA *or an* NFA A is defined in the classical way as the set $L(A)$ of all strings that define a path from one initial state to one of the final states. Two automata A and A' are said to be *equivalent* if they accept the same language, i.e., if $L(A) = L(A')$.

A *strongly connected component* (SCC) C of an NFA or a DFA A is a maximal subset of Q such that, for every two states p and q of C, there exists a path from p to q and a path from q to p in the transition graph of A. Let us denote by $\mathcal{C}_q$ the SCC containing the state $q \in Q$. We consider the partial order $\preceq$ on the set of SCCs of A, defined by $C_1 \preceq C_2$ when 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 $C_1 \neq C_2$.

2.2. *Reversibility*

In this section we introduce the main notions we consider in this paper, by defining different degree of reversibility of automata or of languages.

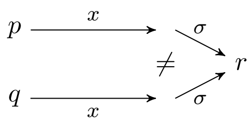
Given a DFA $A = (Q, \Sigma, \delta, q_I, F)$, the *reverse transition function* of A is the function $\delta^{\mathrm{R}} : Q \times \Sigma \to 2^Q$ such that $\delta^{\mathrm{R}}(p, a) = \{q \in Q \mid \delta(q, a) = p\}$. The *reverse automaton* of A is the NFA $A^{\mathrm{R}} = (Q, \Sigma, \delta^{\mathrm{R}}, F, \{q_I\})$ 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 $\#\delta^{\mathrm{R}}(r, a) > 1$ 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 now relax the notion of reversibility, by allowing irreversibility whenever it can be resolved by accessing to a suffix of fixed length of the portion of the input read so far. This yields the notions of *k-reversibility* and *weak-irreversibility.*

Definition 1. *Let k be a positive 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 k-*irreversible* if there exist two states p and q in Q, a string $x \in \Sigma^{k-1}$ and a symbol $\sigma \in \Sigma$, such that $\delta(p, x) \neq \delta(q, x)$ and $\delta(p, x\sigma) = \delta(q, x\sigma) = r$, as depicted here:*



*Otherwise, r is k-*reversible*.*
- *The automaton A is k-*reversible* if each of its states is k-reversible.*
- *The language L is k-reversible if it is accepted by a k-reversible DFA.*
- *The language L is *weakly irreversible* if it is k-reversible for some $k > 0$.*

By definition, a state r is 1-reversible if and only if it is reversible. As a consequence, 1-reversibility (of automata or of languages) 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 σ entering r. However, by knowing the suffix of length k of the part of the input already inspected, i.e., a suffix $x\sigma$ with $|x| = k - 1$, we can uniquely identify which transition on σ 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 access to 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 inspected so far).

Let us denote by REV_k the class of k-reversible languages. As observed previously, $\text{REV} = \text{REV}_1$. Furthermore k-reversible DFAs are denoted REV_kDFAs, for short.

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

Remark 2. *If a state (resp., an automaton or a language) is k-reversible for some positive integer k, then it is k'-reversible for every $k' \geq k$.*

As observed by Pin, not all regular languages are reversible [15]. This result extends to k-reversibility for each $k > 0$ [8]. In particular there exists an infinite

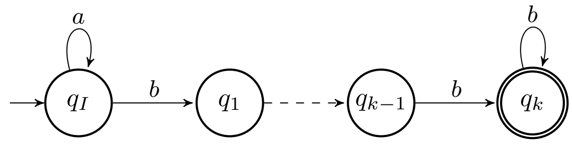


Figure 1: The minimum automaton accepting the language $a^*b^kb^*$

hierarchy of languages, with respect to k-reversibility, as described in the next example.

Example 3 (introduced in [8]) *For each integer $k > 0$, consider the language $L_k = a^*b^kb^*$, 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, namely, the state entered 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., $L_k \in \text{REV}_{k+1} \setminus \text{REV}_k$. This can be also obtained as a consequence of results in Section 5.

As a consequence of Remark 2 and Example 3, we have the proper infinite hierarchy of classes:

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

3. Characterization of k-reversible languages

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 different transitions on the same letter entering in the same state r, where one of them arrives from a state p which belongs to the same strongly connected component of r (see Figure 2a).

We now refine this definition in order to consider strings of the same length that lead to the same state.

Definition 4. *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 $\sigma \in \Sigma$, two strings $x \in \Sigma^{k-1}$ and $w \in \Sigma^*$, such that $\delta(p, x) \neq \delta(q, x)$, $\delta(p, x\sigma) = \delta(q, x\sigma) = r$, and $\delta(r, w) = q$ (see Figure 2b).*

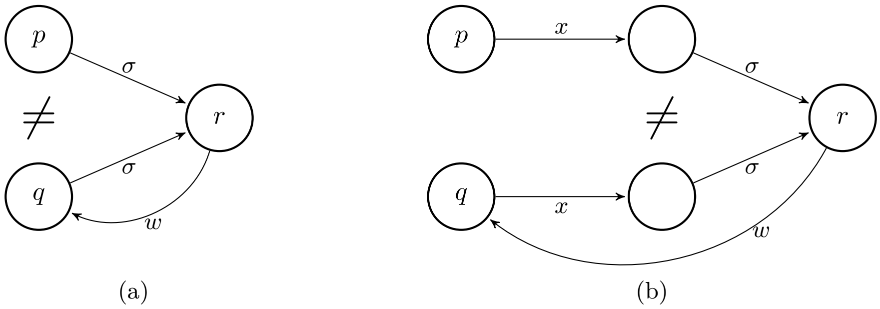


Figure 2: The forbidden pattern (a) and the k-forbidden pattern (b), in which $\sigma \in \Sigma$, $w \in \Sigma^*$, and $x \in \Sigma^{k-1}$

Without the condition $\delta(r, w) = q$, the definition says exactly that r is a k-irreversible state (see first item of Definition 1). With the additional condition, we require that one of the state that "witnesses" the k-irreversibility of r (named q) belongs to same strongly connected component as r, namely $\mathcal{C}_q = \mathcal{C}_r$.

From Definition 4, 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 integer $k' \leq k$.

We will use the notion of k-forbidden pattern 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. In Subsection 3.1, we present an algorithm that transforms any DFA in which does not occur the k-forbidden pattern into an equivalent k-reversible DFA, while in Subsection 3.2, we show that the absence of the k-forbidden pattern in the minimum DFA is a necessary condition for the accepted language to be k-reversible.

3.1. *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 $A' = (Q', \Sigma, \delta', s_I, F')$, which is k-reversible if A does not contain the k-forbidden pattern.

Let us start by presenting an informal outline of the construction. First of all, we discuss the case $k = 1$, namely the case of "standard" reversibility.

If A does not contain the 1-forbidden pattern, to make it reversible we can extend its control to remember the sequence of the states from which, during the computation, irreversible transitions have been executed, i.e., all states $q \in Q$ such that A executed a transition on a symbol σ from q to a state r with $\#\delta^{\mathrm{R}}(r, \sigma) > 1$. With the information saved in this sequence and looking at the previous input symbol, the automaton is able to reverse its computation. Since A does not contain the 1-forbidden pattern, each time it executes an irreversible transition, it moves from an SCC to a different one. This implies that the length of the sequence saved in the control is bounded. Actually, our construction will be slightly redundant: each time A will move from a state q in an SCC to a state in a different SCC we will add q to the sequence, even if the used transition is not irreversible.

The case $k > 1$ is more complicated. Since an irreversible state r could also be entered by an irreversible transition arriving from a state in the SCC $\mathcal{C}_r$, the previous technique would require to remember arbitrarily long sequences of states. We proceed in a different way, still using the previous technique, after the preliminary change in the automaton A that we now illustrate.

For any k-irreversible state r we have the situation in the following picture, for some $p_0, p, q_0, q \in Q$, $a \in \Sigma$, $x \in \Sigma^{k-1}$:

$$\begin{tikzcd}
p_0 \arrow[r, "x"] & p \arrow[dr, "a"] \arrow[d, phantom, "\neq"] & \\
& & r \\
q_0 \arrow[r, "x"'] & q \arrow[ur, "a"'] &
\end{tikzcd}$$

Since A does not contain the k-forbidden pattern, the states p_0 and q_0 cannot belong to $\mathcal{C}_r$. Hence, along each path from p_0 and q_0 to r on xa, there is a transition which, from another SCC, enters $\mathcal{C}_r$. In order to know where those transitions are located, we introduce a counter modulo k in the finite control of A. In the initial configuration the counter contains 0. It is incremented by 1 during the execution of each transition that does not change SCC, while it is set to 0 when the transition reaches a different SCC. In this way, if the state r is reached with 0 in the counter, then both transitions from p and q should arrive from SCCs different than $\mathcal{C}_r$ (otherwise, due to the increment policy of the counter, either p_0 or q_0 should be in $\mathcal{C}_r$). In a similar way, if the counter contains a value ℓ, $0 < \ell < k$, in both paths the transition entering the component $\mathcal{C}_r$ is the one reading the symbol which precedes the suffix of length ℓ of xa, i.e., the one entering the configuration with 0 in the counter.

In this way, if the automaton is in the state r with ℓ in the counter, and it knows the suffix xa of length k of the input read so far and the state which was reached immediately before entering $\mathcal{C}_r$, then it can decide if the state before entering r was p or q and, so, it can reverse the last computation step.

Using these ideas, we now define the DFA A'. The finite control stores three elements:

- The current state q of A.
- An integer $j \in \{0, \ldots, k-1\}$ which is used to count modulo k the visits to states in the current SCC of A, namely in the SCC $\mathcal{C}_q$.
- A sequence of pairs from $Q \times \{0, \ldots, k-1\}$. 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 $\#Q$. It is convenient to view these sequences as words over the alphabet $Q \times \{0, \ldots, k-1\}$, using the symbol $\cdot$ to denote concatenation.

Formally, we give the following definition:

- $Q' = Q \times \{0, \ldots, k-1\} \times (Q \times \{0, \ldots, k-1\})^{<\#Q}$;
- for $\langle q, j, \alpha \rangle \in Q'$, if $\delta(q, a) = p$ then

$$\delta'(\langle q, j, \alpha \rangle, a) = \begin{cases} \langle p, (j+1) \bmod k, \alpha \rangle & \text{if } \mathcal{C}_p = \mathcal{C}_q \\ \langle p, 0, \alpha \cdot (q, j) \rangle & \text{otherwise,} \end{cases}$$

  while $\delta'(\langle q, j, \alpha \rangle, a)$ is left undefined when $\delta(q, a)$ is not defined;
- $s_I = \langle q_I, 0, \varepsilon \rangle$ is the initial state;
- $F' = F \times \{0, \ldots, k-1\} \times (Q \times \{0, \ldots, k-1\})^{<\#Q}$ 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, if $\delta'(\langle p, h, \alpha \rangle, a) = \langle r, \ell, \gamma \rangle$ for some $a \in \Sigma$ and $\langle p, h, \alpha \rangle, \langle r, \ell, \gamma \rangle \in Q'$ with $0 < \ell < k$, then the states p and r are in the same SCC of A and $h = \ell - 1$. This fact will be used in the following proof of the main property of A'.

Lemma 5. *If A does not contain the k-forbidden pattern, then A' is k-reversible.*

Proof. Assume that A does not contain the k-forbidden pattern and, by contradiction, 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 $a \in \Sigma$ and four states $\langle p_0, h_0, \alpha_0 \rangle$, $\langle p, h, \alpha \rangle$, $\langle q_0, j_0\, \beta_0 \rangle$ and $\langle q, j, \beta \rangle$ in Q', such that x defines a path from state $\langle p_0, h_0, \alpha_0 \rangle$ (resp., $\langle q_0, j_0, \beta_0 \rangle$) to state $\langle p, h, \alpha \rangle$ (resp., $\langle q, j, \beta \rangle$), there are transitions on a from both states $\langle p, h, \alpha \rangle$ and $\langle q, j, \beta \rangle$ to state $\langle r, \ell, \gamma \rangle$, and $\langle p, h, \alpha \rangle \neq \langle q, j, \beta \rangle$. The situation is summarized in the following picture:

$$\begin{tikzcd}
\langle p_0, h_0, \alpha_0 \rangle \arrow[r, "x"] & \langle p, h, \alpha \rangle \arrow[dr, "a"] & \\
& \neq & \langle r, \ell, \gamma \rangle \\
\langle q_0, j_0, \beta_0 \rangle \arrow[r, "x"'] & \langle q, j, \beta \rangle \arrow[ur, "a"'] &
\end{tikzcd}$$

First, suppose $\ell = 0$. We divide the proof in the following cases:

Case $\alpha = \beta = \gamma$.

From the definition of δ', $\mathcal{C}_p = \mathcal{C}_q = \mathcal{C}_r$. Furthermore $h = j = k - 1$. Hence $p \neq q$. Since $|x| = k - 1$, in the states of A' along the path from $\langle q_0, j_0, \beta_0 \rangle$ on x, the second component is 0 only at the beginning of the path. This implies that $\mathcal{C}_{q_0} = \mathcal{C}_q = \mathcal{C}_r$. Hence, the automaton A should contain the k-forbidden pattern.

Case $\alpha \neq \gamma$ and $\beta = \gamma$ (and, symmetrically, *Case $\alpha = \gamma$ and $\beta \neq \gamma$*).

In this case $\mathcal{C}_q = \mathcal{C}_r$, while $\mathcal{C}_p \neq \mathcal{C}_r$, which gives $p \neq q$. By the same argument as in the previous case, we obtain $\mathcal{C}_{q_0} = \mathcal{C}_q = \mathcal{C}_r$. So, A should contain the k-forbidden pattern.

Case $\alpha \neq \gamma$ and $\beta \neq \gamma$.
In this case $\mathcal{C}_p \neq \mathcal{C}_r$, $\mathcal{C}_q \neq \mathcal{C}_r$, and $\gamma = \alpha \cdot (p,h) = \beta \cdot (q,j)$. Hence, $\alpha = \beta$, $p = q$, and $h = j$, which contradicts the hypothesis $\langle p,h,\alpha\rangle \neq \langle q,j,\beta\rangle$.

To complete the proof, we are now going to prove that in the case $\ell > 0$ we always obtain a contradiction. By the definition of δ', $h = j = \ell - 1$ and $\alpha = \beta = \gamma$. Hence $p \neq q$. We decompose x as $x'bx''$, where $x', x'' \in \Sigma^*$, $b \in \Sigma$, $|x'| = k - \ell - 1$, and $|x''| = \ell - 1$. Then, for suitable $\langle p_1,h_1,\alpha_1\rangle, \langle q_1,j_1,\beta_1\rangle \in Q'$, $p_2, q_2 \in Q$, we have the following situation:

$$\begin{tikzcd}
\langle p_0,h_0,\alpha_0\rangle \arrow[r, "x'"] & \langle p_1,h_1,\alpha_1\rangle \arrow[r, "b"] & \langle p_2,0,\gamma\rangle \arrow[r, "x''"] & \langle p,\ell-1,\gamma\rangle \arrow[dr, "a"] & \\
 & & & & \langle r,\ell,\gamma\rangle \\
\langle q_0,j_0,\beta_0\rangle \arrow[r, "x'"'] & \langle q_1,j_1,\beta_1\rangle \arrow[r, "b"'] & \langle q_2,0,\gamma\rangle \arrow[r, "x''"'] & \langle q,\ell-1,\gamma\rangle \arrow[ur, "a"'] &
\end{tikzcd}$$

Notice that in the two paths on the string x from $\langle p_0,h_0,\alpha_0\rangle$ to $\langle p,\ell-1,\gamma\rangle$ and from $\langle q_0,j_0,\beta_0\rangle$ to $\langle q,\ell-1,\gamma\rangle$, the last transitions that could change SCC in A are those on the symbol b, immediately after the prefix x'. Suppose that one of these transitions does not change SCC in A, without loss of generality the one from $\langle q_1,j_1,\beta_1\rangle$ to $\langle q_2,0,\gamma\rangle$. Then, by the definition of δ', $j_1 = k-1$, and, since $|x'| < k$, none of the second components of the states on the path from $\langle q_0,j_0,\beta_0\rangle$ to $\langle q_1,j_1,\beta_1\rangle$ on x' can be 0. This implies that $\mathcal{C}_{q_0} = \mathcal{C}_{q_1} = \mathcal{C}_{q_2} = \mathcal{C}_q = \mathcal{C}_r$. Therefore, A should contain the k-forbidden pattern, which is a contradiction. Thus, in A both transitions on b from p_1 to p_2 and from q_1 to q_2 should change SCC. Considering the corresponding transitions in the two paths under consideration, we get that $\gamma = \alpha_1 \cdot (p_1,h_1) = \beta_1 \cdot (q_1,j_1)$ and, hence, $p_1 = q_1$. Since A is deterministic, this implies $p_2 = q_2$ and, finally, $p = q$, namely another contradiction. □

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

Theorem 6. *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 an n-state DFA not containing the k-forbidden pattern. According to Lemma 5, the automaton A' obtained from A with the above presented construction is k-reversible. We now estimate the number of its reachable states.

First, notice that if a state $\langle q,\ell,\alpha\rangle$ with $\alpha = (p_1,j_1)(p_2,j_2)\cdots(p_h,j_h)$ is reachable, then $\mathcal{C}_{p_1} \prec \mathcal{C}_{p_2} \prec \ldots \prec \mathcal{C}_{p_h} \prec \mathcal{C}_q$. Hence, since the ordering of the pairs appearing in α (i.e., the letters of 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

$f : Q \to \{-, 0, 1, \ldots, k-1\}$, such that for $r \in Q$:

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

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 in A all the transitions entering any state in the SCC of the initial state q_I can arrive only from states in the same SCC. Thus, if some state in $\mathcal{C}_{q_I}$ is k-irreversible, then A should contain the k-forbidden pattern, which is a contradiction. This allows to directly simulate all the states in $\mathcal{C}_{q_I}$, without using the counter. As a consequence, in each state $\langle q, \ell, \alpha \rangle$ of Q', with $q \notin \mathcal{C}_{q_I}$, the first element of α, which should represent a state in $\mathcal{C}_{q_I}$, is stored without the counter. Hence, such a state can be seen as a pair whose first component is a state in $\mathcal{C}_{q_I}$ (the first element of α) and the second component is the above function f restricted to the set $Q \setminus \mathcal{C}_{q_I}$ (representing both the current state q with its counter ℓ and the other pairs in α). Since $f(q) = \ell \in \{0, \ldots, k-1\}$, f cannot be the constant function $f(r) = -$ for $r \in Q \setminus \mathcal{C}_{q_I}$. Hence, the number of possible functions is bounded by $(k+1)^{n-s} - 1$, where $s = \#\mathcal{C}_{q_I}$. Considering also the states which are used in Q' to simulate the states in $\mathcal{C}_{q_I}$, this gives at most $s + s\left((k+1)^{n-s} - 1\right)$ many reachable states. For $k > 0$ this amount is bounded by $(k+1)^{n-1}$. $\square$

We point out that for $k = 1$, Theorem 6 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.

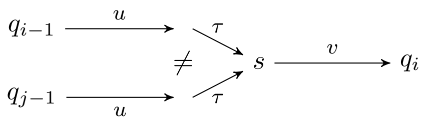
3.2. *The characterization*

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 6 with the following result.

Lemma 7. *Let L be a regular language and k be a positive integer. If the minimum DFA accepting L contains the k-forbidden pattern, then $L \notin \text{REV}_k$.*

Proof. Let $M = (Q, \Sigma, \delta, q_I, F)$ be the minimum DFA accepting L. By hypothesis there exist three states $p, q, r \in Q$, a symbol $\sigma \in \Sigma$, and two strings $x \in \Sigma^{k-1}$ and $w \in \Sigma^*$ such that $\delta(p, x) \neq \delta(q, x)$, $\delta(p, x\sigma) = \delta(q, x\sigma) = r$ and $\delta(r, w) = q$ (see Figure 2b). Let $A' = (Q', \Sigma, \delta', s_I, F')$ be a DFA accepting L. We are going to prove that A' contains a k-irreversible state.

Let $q_0 \in Q'$ be a state equivalent to p. Because A' is equivalent to M, for every $n \in \mathbb{N}$, the state $q_n = \delta'(q_0, (x\sigma w)^n)$ is equivalent to q. Since Q' is finite, sooner or later, some repetition occur in the so-defined sequence of q_i's. Let i and j be the first two indices such that $q_i = q_j$ with $1 \leq i < j$ (i cannot be equal to 0, because q_j is equivalent to q while q_0 is equivalent to p). Now, consider the two paths from q_{i-1} and q_{j-1} to q_i on string $x\sigma w$. By minimality of i and j, $q_{i-1} \neq q_{j-1}$. Hence, we can decompose the string $x\sigma w$ as $u\tau v$ for some $u, v \in \Sigma^*$ and $\tau \in \Sigma$, such that $\delta'(q_{i-1}, u) \neq \delta'(q_{j-1}, u)$ and $\delta'(q_{i-1}, u\tau) = \delta'(q_{j-1}, u\tau) = s$ for some state $s \in Q'$. The situation is depicted here:



In order to show that s is k-irreversible we proceed in two cases.

Case $i = 1$. Since $q_{i-1} = q_0$ is equivalent to p and q_{j-1} is equivalent to q, we get that $\delta'(q_{i-1}, x)$ is equivalent to $\delta(p, x)$ which is different from $\delta(q, x)$. It follows that $\delta'(q_{i-1}, x) \neq \delta'(q_{j-1}, x)$ and, since x is a prefix of u, $|u| \geq |x| = k - 1$. Thus, s is k-irreversible by Remark 2.

Case $i > 1$. We directly obtain that $\delta(q_{i-2}, x\sigma wu) \neq \delta(q_{j-2}, x\sigma wu)$, while $\delta(q_{i-2}, x\sigma wu\tau) = \delta(q_{j-2}, x\sigma wu\tau) = s$. Hence, s is $|x\sigma wu|$-irreversible and finally k-irreversible by Remark 2. □

An immediate consequence of Lemma 7 and Theorem 6, which strengthens the statement of Lemma 7, is that whenever a minimum automaton contains the k-forbidden pattern, so does any equivalent DFA. However, the converse is not true in general. The condition in Lemma 7 is indeed 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).

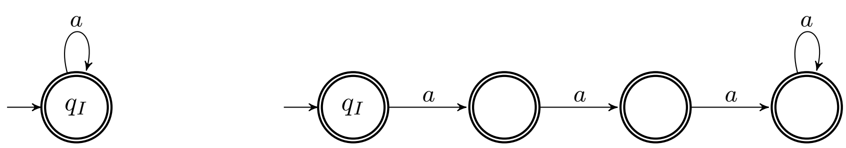


Figure 3: The minimum DFA accepting the reversible language a^*, and an equivalent DFA containing the 3-forbidden pattern

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

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 6, while the *only-if* part derives from Lemma 7. □

From Theorem 8, we observe that to transform each DFA A accepting a k-reversible language into an equivalent REV_kDFA, firstly we can transform A into the equivalent minimum DFA M and then we can apply to M the construction presented in Subsection 3.1.

As a consequence of Theorem 8, we also obtain:

Corollary 9. *$L \in \text{REV}_{k+1} \setminus \text{REV}_k$ if and only if the maximum h such that the minimum* DFA *accepting L contains the h-forbidden pattern is k.*

4. 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*, which we define now.

Definition 10. *Let k be a positive 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* strongly irreversible *if it is k-irreversible for each $k > 0$.*
- *The automaton A is* strongly irreversible *if it contains at least one strongly irreversible state.*
- *The language L is* strongly irreversible *if it is k-irreversible for each $k > 0$, i.e., if L is not weakly irreversible.*

First of all, we observe that, by Theorem 8, 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 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. This "large enough" forbidden pattern will actually be proved to be equivalent to the *strong forbidden pattern* that we introduce now.

Definition 11. *Given a* DFA *$A = (Q, \Sigma, \delta, q_I, F)$, the* strong forbidden pattern *is formed by three states p, q and r in Q, three strings $v \in \Sigma^+$, $x, w \in \Sigma^*$ and a symbol $\sigma \in \Sigma$ such that $\delta(q, v) = q$, $\delta(p, v) = p$, i.e., v defines self-loops over both p and q, $\delta(p, x) \neq \delta(q, x)$, $\delta(p, x\sigma) = \delta(q, x\sigma) = r$ and $\delta(r, w) = p$ (see Figure 4).*

The next result states the equivalence of three conditions that allows us to test whether an automaton contains the k-forbidden pattern for each $k > 0$.

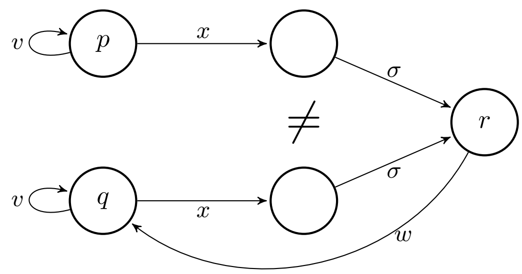


Figure 4: The strong forbidden pattern: $v \in \Sigma^+$, $x \in \Sigma^*$, $\sigma \in \Sigma$, $w \in \Sigma^*$

Theorem 12. *Let A be an n-state* DFA *and $N = \frac{n^2-n}{2}$. The three following statements are equivalent:*

(1) A contains the $(N+1)$-forbidden pattern;
(2) A contains the strong forbidden pattern;
(3) A contains the k-forbidden pattern for each $k > 0$.

Proof.

(1) $\implies$ (2) Suppose now that A contains the $(N+1)$-forbidden pattern. Then, there exist $p, q, r \in Q$, $\sigma \in \Sigma$, and $x \in \Sigma^N$, such that $\delta(p, x) \neq \delta(q, x)$ while $\delta(p, x\sigma) = \delta(q, x\sigma) = r$ and $\mathcal{C}_r = \mathcal{C}_q$. Let $x = \sigma_1\sigma_2 \cdots \sigma_N$ with $\sigma_i \in \Sigma$ for $i = 1, \ldots, N$ and let $p_0, \ldots, p_N$ (resp., $q_0, \ldots, q_N$) be the sequence of states entered from p (resp., q) when successively reading the symbols σ_i's, i.e., $p = p_0$ (resp., $q = q_0$) and $p_i = \delta(p_{i-1}, \sigma_i)$ (resp., $q_i = \delta(q_{i-1}, \sigma_i)$) for $i = 1, \ldots, N$. We furthermore have $\delta(p_N, \sigma) = \delta(q_N, \sigma) = r$. Because $\mathcal{C}_{q_0} = \mathcal{C}_r$, we have $\mathcal{C}_{q_i} = \mathcal{C}_r$ for $i = 0, \ldots, N$. Moreover, since $p_N \neq q_N$ and A is deterministic, we get $p_i \neq q_i$ for $i = 0, \ldots, N$. We consider the pairs $(p_0, q_0), \ldots, (p_N, q_N)$. Notice that there are $n^2 - n$ possible pairs of different states. Thus, there exist two indices i and j, $0 \leq i < j \leq N$, such that either $(p_i, q_i) = (p_j, q_j)$ or $(p_i, q_i) = (q_j, p_j)$. In both cases, $\delta(p_i, (\sigma_{i+1} \cdots \sigma_j)^2) = p_i$ and $\delta(q_i, (\sigma_{i+1} \cdots \sigma_j)^2) = q_i$. This directly gives a strong forbidden pattern with states p_i, q_i and r and self-loops around p_i and q_i on the nonempty string $(\sigma_{i+1} \cdots \sigma_j)^2$.

(2) $\implies$ (3) Suppose that A contains the strong forbidden pattern, and fix a positive integer k. By definition, there exist $p, q, r \in Q$, $v \in \Sigma^+$, $x \in \Sigma^*$ and $\sigma \in \Sigma$ such that v defines self-loops around both p and q, $\delta(p, x) \neq \delta(q, x)$ while $\delta(p, x\sigma) = \delta(q, x\sigma) = r$ and $\mathcal{C}_r = \mathcal{C}_q$. Then, by setting $x' = v^k x$ we obtain a $|x'|$-forbidden pattern using the same states p, q and r, the string x' and the symbol σ. Since $|x'| \geq k$, we conclude using Remark 2 that A contains the k-forbidden pattern.

(3) $\implies$ (1) Immediate.

□

Combining Theorem 8 with Theorem 12 we obtain:

Corollary 13. *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 14. *The language $L = a^*b(a+b)^*$ is strongly irreversible. The minimum automaton accepting it has 2 states (see Figure 5). We notice that $\delta(q_I, a) \neq \delta(p, a)$, while $\delta(q_I, ab) = \delta(p, ab) = p$. This defines a 2-forbidden pattern. According to Corollary 13, this implies that L is strongly irreversible. Observe that since the string a defines two self-loops around both q_I and p, we actually have a strong forbidden pattern, and therefore a k-forbidden pattern for any $k > 0$ using the string $a^k b$.*

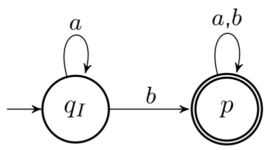


Figure 5: The minimum automaton accepting the language $L = a^*b(a+b)^*$

5. Degree of Reversibility: Automata versus Languages

In the following result we present further families of languages, besides that in Example 3, which witness the existence of the proper infinite hierarchy

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

Furthermore, we show that the difference between the "amount" of irreversibility in a minimum DFA and in the accepted language can be arbitrarily large, or even infinite. For $k \in \mathbb{N}$, we say that an automaton A (resp., a language L) has *degree of irreversibility* k if it is $(k+1)$-reversible but not k-reversible, namely, if k is the maximum integer such that A is not a REV_kDFA (resp., L does not belong to REV_k). If A (resp., L) is strongly irreversible, then we say that it has *degree of irreversibility* ∞.

Theorem 15. *For all $k, j \in \mathbb{N} \cup \{\infty\}$ with $0 < k \leq j$, there exists a regular language $L_{k,j}$ such that:*

- *the minimum* DFA *accepting $L_{k,j}$ has degree of irreversibility j;*
- *$L_{k,j}$ has degree of irreversibility k.*

Proof. We first consider the case $k, j \in \mathbb{N}$. When $k = j$ it is enough to consider the minimum DFA accepting the language $a^*b^kb^*$ (cf. Example 3), which is $(k+1)$-reversible but contains the k-forbidden pattern.

From now on, we suppose $k < j$.

Let $L_{k,j}$ be the language accepted by the automaton $A_{k,j} = (Q, \Sigma, \delta, q_I, F)$ where $\Sigma = \{a, b\}$, $Q = \{q_I, r, q_F\} \cup \{s_1, \ldots, s_j\} \cup \{t_1, \ldots, t_j\}$, $F = \{t_j, q_F\}$, and the transition function is defined as follows (see Figure 6 for an example):

- $\delta(q_I, a) = s_1$ and $\delta(q_I, b) = t_1$
- $\delta(s_i, a) = s_{i+1}$ and $\delta(t_i, a) = t_{i+1}$ for $1 \leq i \leq j - k$
- $\delta(s_i, b) = s_{i+1}$ and $\delta(t_i, b) = t_{i+1}$ for $j - k < i < j$
- $\delta(s_{j-1}, b) = \delta(t_{j-1}, b) = \delta(r, b) = r$
- $\delta(r, a) = q_F$

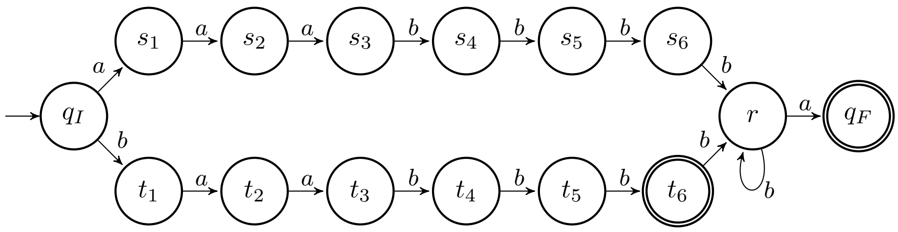


Figure 6: The minimum automaton $A_{4,6}$ accepting the language $L_{4,6}$

Firstly, we can observe that $A_{k,j}$ is a minimum DFA. It contains only one irreversible state, namely r, with $\delta^{\mathrm{R}}(r, b) = \{r, s_j, t_j\}$. We also notice that $\delta(s_1, a^{j-k}b^{k-1}) = s_j$ is different from $\delta(t_1, a^{j-k}b^{k-1}) = t_j$, while $\delta(s_1, a^{j-k}b^k) = \delta(t_1, a^{j-k}b^k) = r$. Hence $A_{k,j}$ is not a REV_jDFA. However, the knowledge of one more symbol in the suffix of the input read to enter r allows to determine the state of the automaton before reading the last symbol. In particular, it is s_j (resp., t_j) if the suffix of length j is $a^{j-k+1}b^k$ (resp., $ba^{j-k}b^k$). Hence, $A_{k,j}$ is a REV_{j+1}DFA .

In order to prove that $L_{k,j}$ has degree of irreversibility k, we first show that $A_{k,j}$ contains the k-forbidden pattern. Indeed, denoting by p the state s_{j-k+1} and by x the string b^{k-1}, we observe that $\delta(p, x) = s_j$ is different from $\delta(r, x) = r$, while $\delta(p, xb) = \delta(r, xb) = r$ (in this case, the states q and r from Definition 4 are equal and therefore in the same SCC). Furthermore, it is possible to obtain a REV_{k+1}DFA $A'_{k,j}$ equivalent to $A_{k,j}$ by duplicating r and q_F with their transitions and by redistributing the incoming transitions from s_j and t_j, as in the case presented in Figure 7. Hence, L_k, j has degree of irreversibility k.

We now consider the case $j = \infty$. If $k = \infty$, it is enough to consider a strongly irreversible language and its minimum automaton, e.g., the language $a^*b(a + b)^*$ given in Example 14. Otherwise, $k \in \mathbb{N}$. Let us first consider the case $k = 1$. We

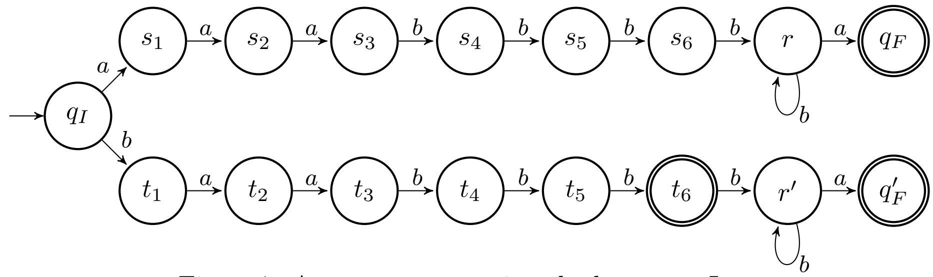


Figure 7: A REV_5DFA accepting the language $L_{4,6}$

define the language $L_{1,\infty} \subseteq \Sigma^*$ by describing its minimum automaton $A_{1,\infty}$ (see Figure 8). The automaton uses three states q_I, which is the initial state, p and q_F which are the final states. There are transitions on b from q_I to p and from p to q_I, and transitions on a from both p and q_I to r. Therefore, r is irreversible. It is in fact strongly irreversible, since the string $b^n a$ defines paths from both q_I and p to r, for any $n \in \mathbb{N}$. However, it does not contain the forbidden pattern (see Figure 2a), and, thus, the accepted language is reversible. To obtain the languages $L_{k,\infty}$ for $k > 1$, it

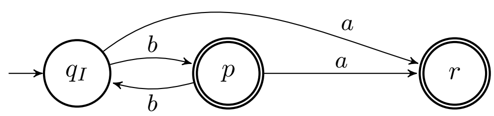


Figure 8: The strongly irreversible minimum DFA $A_{1,\infty}$ accepting the reversible language $L_{1,\infty}$

is enough to consider the concatenation of the above-defined language $L_{k,j}$ for some arbitrary $j > k$ and the language $L_{1,\infty}$. Indeed, the minimum automaton (consisting of the minimum automaton accepting $L_{k,j}$ presented above, in which the state q_F is replaced by the entire automaton $L_{1,\infty}$[a]) is strongly irreversible (due to the second part), but the language has degree of irreversibility k (due to the first part). □

6. 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, namely, the degree of irreversibility of L.

The idea is to simultaneously analyze all the paths entering each irreversible state $r \in Q$ of the minimum DFA A accepting L in order to find the

[a]In this case, this "concatenation" of automata does not create additional irreversibility, since in $A_{k,j}$, the state q_F is entered only by transitions with label a, while in $A_{1,\infty}$, the state q_I is entered only by transitions with label b.

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., $A^{\mathrm{R}} \times A^{\mathrm{R}}$, 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)reversibility of L — not of A — it is useful to recall that by Definition 4 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., $\mathcal{C}_r = \mathcal{C}_q$. 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 a DFA, $A^{\mathrm{R}} = (Q, \Sigma, \delta^{\mathrm{R}}, F, \{q_I\})$ be the reversal automaton of A, and $A^{\mathrm{R}}_{\mathrm{SCC}s} = (Q, \Sigma, \delta^{\mathrm{R}}_{\mathrm{SCC}s}, F, \{q_I\})$ be the NFA obtained by splitting A^{R} in its SCCs, i.e., $\delta^{\mathrm{R}}_{\mathrm{SCC}s}(r, a) = \{q \mid q \in \delta^{\mathrm{R}}(r, a) \text{ and } \mathcal{C}_r = \mathcal{C}_q\}$ for $r \in Q$ and $a \in \Sigma$. Let us define the nondeterministic automaton $\hat{A}$ as follows: $\hat{A} = (\hat{Q}, \Sigma, \hat{\delta}, \hat{I}, \hat{F})$ where

- $\hat{Q} = Q \times Q$;
- $\hat{I} = \{(r, r) \mid r \in Q\}$;
- $\hat{F} = \hat{Q} \setminus \hat{I}$;
- $\hat{\delta}((r', r''), a) = \{(p, q) \in \delta^{\mathrm{R}}(r', a) \times \delta^{\mathrm{R}}_{\mathrm{SCC}s}(r'', a) \mid p \neq q\}$.

So defined, the resulting automaton $\hat{A}$ accepts the strings z such that z^{R} defines a $|z|$-forbidden pattern in A. Formally, this follows from the following lemma:

Lemma 16. *Given a string x and a symbol σ, in the DFA A, three states p, q and r with x, σ and $\mathcal{C}_q = \mathcal{C}_r$ form a $|x\sigma|$-forbidden pattern if and only if in $\hat{A}$ there exists a path on σx^{R} from the state (r, r) to the state (p, q), i.e., $(p, q) \in \hat{\delta}\,((r, r), \sigma x^{\mathrm{R}})$.*

Proof. (only if) Since $\delta(p, x\sigma) = \delta(q, x\sigma) = r$ and $\mathcal{C}_q = \mathcal{C}_r$, we have $p \in \delta^{\mathrm{R}}(r, \sigma x^{\mathrm{R}})$ and $q \in \delta^{\mathrm{R}}_{\mathrm{SCC}s}(r, \sigma x^{\mathrm{R}})$. Moreover, the sequences $p_0, p_1, \ldots, p_k$ and $q_0, q_1, \ldots, q_k$ of states visited in A along the paths defined by $x\sigma$ respectively from $p = p_0$ to $r = p_k$ and from $q = q_0$ to $r = q_k$, satisfy $p_i \neq q_i$ for each $i < k$. Hence, by definition of $\hat{\delta}$, there exists a path from (r, r) to (p, q) on σx^{R} in $\hat{A}$, i.e., $(p, q) \in \hat{\delta}\,((r, r), \sigma x^{\mathrm{R}})$.

(if) For the converse implication, we proceed by induction on the length of x.

(Basis) If $|x| = 0$, namely $x = \varepsilon$, then $(p, q) \in \hat{\delta}\,((r, r), \sigma)$. By definition of $\hat{\delta}$, it follows that $\delta(p, \sigma) = \delta(q, \sigma) = r$ while $p \neq q$ and $\mathcal{C}_r = \mathcal{C}_q$, namely, p, q and r form a 1-forbidden pattern using σ.

(Induction step) If $|x| > 0$, then $x = \tau y$ where τ is a symbol and $|y| = k - 1$. Assume that there exists a path from (r, r) to (p, q) on the string $\sigma x^{\mathrm{R}} = \sigma y^{\mathrm{R}} \tau$. Then, there is a path on σy^{R} from (r, r) to some state (p', q') such that $(p, q) \in \hat{\delta}\,((p', q'), \tau)$. Again, by definition of $\hat{\delta}$, we obtain $\delta(q, \tau) = q'$, $\delta(p, \tau) = p'$ and $\mathcal{C}_{q'} = \mathcal{C}_q$. By the induction hypothesis, we get that p', q' and r with y and σ form a $|y\sigma|$-forbidden pattern in A. Since $\mathcal{C}_r = \mathcal{C}_{q'} = \mathcal{C}_q$, we finally obtain that p, q and r with x and σ form a $|x\sigma|$-forbidden pattern in A. □

Considering Theorem 8, this leads to state the following:

Lemma 17. *Let A be a minimum n-state* DFA *and $\hat{A}$ be the above-described* NFA *defined from A. Then:*

- *The following statements are equivalent:*
  - $L(A)$ *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 $k > 0$, $L(A) \in \text{REV}_k$ if and only if $L(\hat{A})$ contains only strings of length less than k.*

Proof. The first item directly follows from Theorem 12, Corollary 13, and Lemma 16. The second item follows from Corollary 9 and Lemma 16. □

The same argument can be exploited to prove that the problem of deciding 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 18. *The problem of deciding whether a language is strongly or weakly irreversible is NL-complete.*

Proof. From the minimum DFA accepting the language under consideration we derive 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 (see first item of Lemma 17). 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, i.e., G belongs to GAP.

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

i. $\delta(i, j) = j$, for $(i, j) \in E$
ii. $\delta(0, 1) = \delta(n, 1) = 1$
iii. $\delta(0, \$) = 0$ and $\delta(n, \$) = n$
iv. $\delta(q_I, i) = i$, for $i = 0, \ldots, n$
v. $\delta(i, \sharp) = \sharp$, for $i = 1, \ldots, n$

Notice that the restriction of the underlying graph of A to states $1, \ldots, n$ coincides with G (transitions i), plus the edge from n to 1 (from transitions ii), if it does

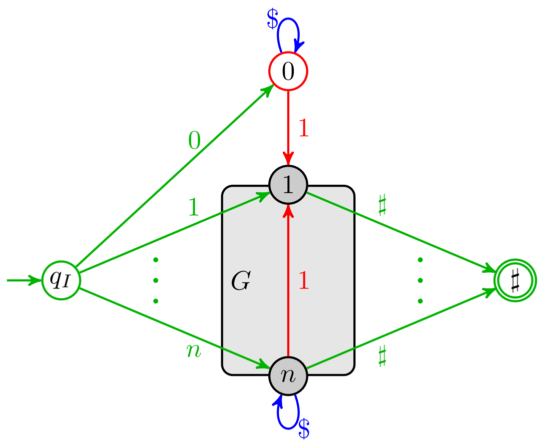


Figure 9: The automaton A obtain from a graph G. Only transitions that have been added to G (i.e., transitions ii to v) are depicted

not already exists in G[b]. Furthermore, to states 0 and n we added a self-loop on the symbol \$ (transitions iii) and a transition on symbol 1 going to state 1 (transitions ii), in order to make the state 1 strongly irreversible. Lastly, we made each state useful (transitions iv enforce reachability while transitions v enforce productivity of each state).

By observing that, for $\sigma \neq \$$, any transition labeled by σ enters the state σ, we can prove that \$1 is the unique string of length 2 that induces irreversibility, and this 2-irreversibility occurs in state 1. Hence, there are no states other than 1 which are strongly irreversible. Thus, we obtain that the strong forbidden pattern occurs in A only if it is formed by the states 0, n and 1. This is possible only if the states 1 and n are in the same SCC because, by construction, there is no path from 1 to 0. Observe that transitions ii already ensures that a path from n to 1 exists. Hence, we can conclude that $L(A)$ is strongly irreversible if and only if there exists a path from 1 to n in G.

This reduction can be computed in deterministic logarithmic space. □

7. Conclusion

We introduced and studied the notions of strong and weak irreversibility for finite automata and regular languages. As we have seen in Section 5 (see also [8]), there exists an infinite hierarchy of weakly irreversible languages, which is itself strictly included in the class of regular languages by the existence of strongly irreversible

[b]It may happen that the transition from state n to state 1 on symbol 1 (from transitions ii) is already defined as being an edge in G (transitions i); we nevertheless ensure that the transition exists in A.

languages as proved in Section 4. In both cases, the witness languages are defined over a binary alphabet, so the question arises whether 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 length less than k states. Hence, for each k, the language $a^k a^*$ is $(k+1)$-reversible, but not k-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 positive integer k. 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.

Bibliography

[1] D. Angluin, Inference of reversible languages, *J. ACM* **29**(3) (1982) 741–765.

[2] C. H. Bennett, Logical reversibility of computation, *IBM Journal of Research and Development* **17**(6) (1973) 525–532.

[3] M. Holzer, S. Jakobi and M. Kutrib, Minimal reversible deterministic finite automata, *International Journal of Foundations of Computer Science* **29**(2) (2018) 251–270.

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

[5] N. D. Jones, Space-bounded reducibility among combinatorial problems, *Journal of Computer and System Sciences* **11**(1) (1975) 68–85.

[6] A. Kondacs and J. Watrous, On the power of quantum finite state automata, *FOCS*, (IEEE Computer Society, 1997), pp. 66–75.

[7] M. Kutrib, Reversible and irreversible computations of deterministic finite-state devices, *Mathematical Foundations of Computer Science 2015 - 40th International Symposium, MFCS 2015, Milan, Italy, August 24-28, 2015, Proceedings, Part I*, eds. G. F. Italiano, G. Pighizzini and D. Sannella *Lecture Notes in Computer Science* **9234**, (Springer, 2015), pp. 38–52.

[8] M. Kutrib and T. Worsch, Degrees of reversibility for DFA and DPDA, *Reversible Computation - 6th International Conference, RC 2014, Kyoto, Japan, July 10-11, 2014. Proceedings*, eds. S. Yamashita and S. Minato *Lecture Notes in Computer Science* **8507**, (Springer, 2014), pp. 40–53.

[9] R. W. Landauer, Irreversibility and heat generation in the computing process, *IBM Journal of Research and Development* **5** (July 1961) 183–191.

[10] K. Lange, P. McKenzie and A. Tapp, Reversible space equals deterministic space, *Journal of Computer and System Science* **60**(2) (2000) 354–367.

[11] G. J. Lavado, G. Pighizzini and L. Prigioniero, Minimal and reduced reversible automata, *Journal of Automata, Languages and Combinatorics* **22**(1-3) (2017) 145–168.

[12] G. J. Lavado and L. Prigioniero, Concise representations of reversible automata, *International Journal of Foundations of Computer Science* **30**(6-7) (2019) 1157–1175.

[13] S. Lombardy, On the construction of reversible automata for reversible languages, *Automata, Languages and Programming, 29th International Colloquium, ICALP 2002, Malaga, Spain, July 8-13, 2002, Proceedings*, eds. P. Widmayer, F. T. Ruiz, R. M. Bueno, M. Hennessy, S. Eidenbenz and R. Conejo *Lecture Notes in Computer Science* **2380**, (Springer, 2002), pp. 170–182.

[14] G. Pighizzini, Investigations on automata and languages over a unary alphabet, *International Journal of Foundations of Computer Science* **26**(7) (2015) 827–850.

[15] J. Pin, On reversible automata, *LATIN '92, 1st Latin American Symposium on Theoretical Informatics, São Paulo, Brazil, April 6-10, 1992, Proceedings*, ed. I. Simon *Lecture Notes in Computer Science* **583**, (Springer, 1992), pp. 401–416.