Федеральное государственное автономное образовательное учреждение высшего образования

**«Уральский Федеральный Университет<br>имени первого Президента России Б.Н.Ельцина»**

Институт естественных наук и математики

Кафедра алгебры и фундаментальной информатики

На правах рукописи

**Давид Фернандо Касас Торрес**

# ИССЛЕДОВАНИЯ ВПОЛНЕ ДОСТИЖИМЫХ АВТОМАТОВ

1.2.3 Теоретическая информатика, кибернетика

Диссертация на соискание учёной степени
кандидата физико-математических наук

Научный руководитель
доктор физико-математических наук
профессор Волков Михаил Владимирович

Екатеринбург, 2024

URAL FEDERAL UNIVERSITY
INSTITUTE OF NATURAL SCIENCE AND MATHEMATICS

David Fernando Casas Torres

# INVESTIGATIONS ON COMPLETELY REACHABLE AUTOMATA

1.2.3 — Theoretical Computer Science, Cybernetics

A Thesis Submitted for the Degree
of Candidate of Physical and Mathematical Sciences

Supervisor: Doctor
of Physical and Mathematical
Sciences, Professor M. V. Volkov

Ekaterinburg, 2024

# Contents

# Introduction

## Relevance of the research

Deterministic Finite Automata (DFAs) are simple but very versatile theoretical devices. Their simplicity does not hinder them from being a very interesting field of research for discrete mathematics and computer science alike. One of their most widespread applications is as language recognizers. Thanks to Kleene's classical theorem it is long known that DFAs are able to recognize the class of regular languages. In this text this aspect will not be studied. Nevertheless the interest in DFAs is by no means limited to this. They can represent how a closed but mutable system changes in the presence of some inputs. DFAs are studied as (theoretical) machines that constantly change internal state (and probably output) depending on the different inputs they receive. They can model various real life systems: from vending machines to simple artificial intelligences.

For a better discussion, let us present some informal definitions of key concepts treated in this work (bear in mind that more detailed definitions will be made in the following sections.) A DFA, or from now on an automaton, is usually defined as a triple $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$ where $Q$ and $\Sigma$ are finite sets called the *state set* and *alphabet* respectively and $\delta$ is a function from $Q \times \Sigma$ to $Q$. The set $Q$ represents the possible internal states in which the automaton can be. The letters of $\Sigma$ represent all the valid inputs received by the automaton. The function $\delta$ maps all the possible situations, pairs of state and input, the automaton could be in to the respective output represented as a state. An automaton can be considered as a machine that receives some inputs, the letters, and depending on the state it is at the moment it will change to another (not necessarily different) state. In order to simplify the henceforth discussion, we use the notation *to the right*, i.e., for the pair $(q, a) \in Q \times \Sigma$ we represent $\delta(q, a)$ as $q \cdot a$. This allows us to omit the reference to the function $\delta$ when we refer to automata, and hence just consider the state set and the alphabet, i.e., we write $\mathcal{A} = \langle Q, \Sigma \rangle$. Additionally, we can concatenate multiple inputs in a *word*, a finite sequence of letters, and define its action at

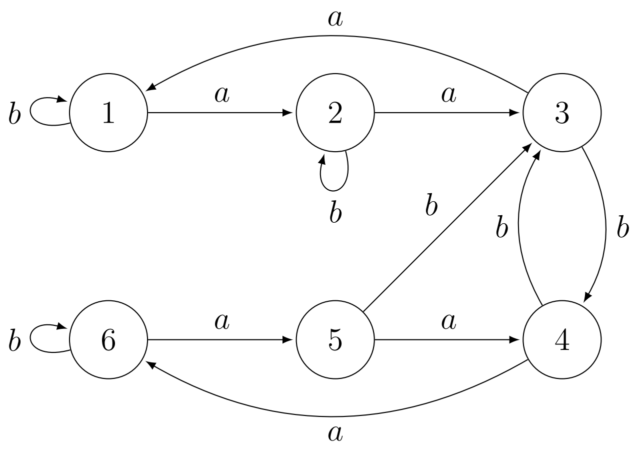


Figure 1: The labelled directed graph of an automaton.

any state. If $w = a_1 a_2 \cdots a_n$, we denote by $q \cdot w$ the state at the end of the sequential action of $a_1$ followed by the action of $a_2$ and so on until reaching $a_n$. As it is usual in the literature, $\Sigma^*$ represents the set of all the words that can be formed using letters of the alphabet $\Sigma$. Similar to how we can apply multiple letters to a single state, we can apply any letter $a$ to a non-empty subset of states $P \subset Q$. The subset of states obtained by this operation is denoted by $P \cdot a$. At this point nothing prevents us to apply a word $w$ to a subset of states $P$ and obtain $P \cdot w$.

One of the most useful representations of automata is as a labelled directed graph. The vertices of this graph are the states and the directed edges are labelled by the letters of its alphabet representing their actions on the states. More precisely, if $\mathcal{A} = \langle Q, \Sigma \rangle$ is an automaton, in the graph representation of $\mathcal{A}$ there is an edge that connects states $p, q \in Q$, labelled by the letter $a$, i.e., $p \overset{a}{\to} q$, if and only if $p \cdot a = q$. Figure 1 shows the graph representation of an automaton with six states and two letters. Note that the presence of a loop, at a state, labelled by $a$ represents that the action of the letter does not change the state.

## Synchronization

Every physical system is susceptible to errors or temporal disconnections that would make the user to lose track of the system's current state. Because of this it would be convenient to have a sequence of inputs that, without regard of the current situation of the system, once finished the user could know with total certainty in which state the system is. This is one of the different motivations of the notion of *synchronization* of automata. In plain words,

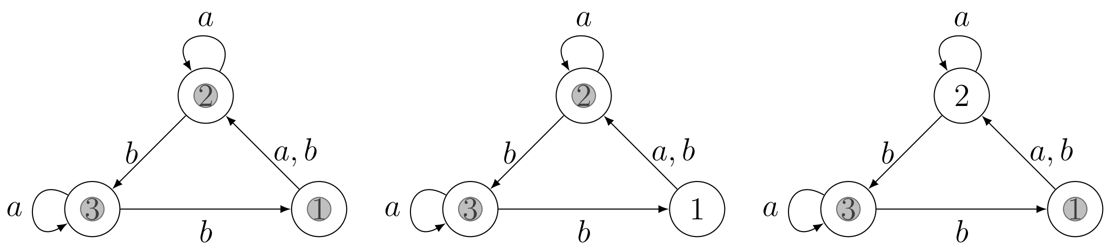


Figure 2: The representation of an automaton with tokens and the effect over this tokens of the empty word, $a$ and $ab$ respectively.

a *synchronizable automaton* allows one to have an input that, no matter in which state the machine is, will always end in one a priori known state. The multiple times that this concept was conceived independently through history is an evidence of how natural it is. At the same time its ubiquity shows how useful it can be. We give a brief historic discussion not much later in this section.

Before, let us show a helpful graphic representation of the concept of synchronization. Given an automaton $\mathcal{A} = \langle Q, \Sigma \rangle$ and its graph, there is a mental image of the effect of a word over a state. It goes as follows: let $q \in Q$ be an arbitrary state. Imagine a token over the vertex $q$ in the directed graph of $\mathcal{A}$ and let $w = a_1 \cdots a_m \in \Sigma^*$ be a word with $a_i \in \Sigma$ for every $i = 1, \ldots, m$. At first move the token from $q$ to the vertex $q \cdot a_1$, and then to the state $(q \cdot a_1) \cdot a_2$ and so on, following the edges indicated by the word. At the end it is easy to see to which state the word $w$ sends the state $q$. With this in mind now imagine the same situation with tokens on every state of a nonempty subset $P \subseteq Q$ and a word $w \in \Sigma^*$. Follow the previously described procedure, but in the case two or more tokens fall in the same vertex at the same time remove all but one of the tokens in that vertex and continue the process. In this way the image of $P$ by the word $w$, i.e. $P \cdot w$, is the set of states with tokens at the end of the process. In Figure 2 we can see a graphic representation of this. The leftmost automaton has tokens in each of its states; the center automaton represents the situation after applying the letter $a$; and the rightmost represents the position of the tokens after the word $ab$. If it happens that once finished the procedure there is just one token left, then it is said that the word $w$ *synchronizes* the set $P$. Suppose that in the previous situation $P$, is the whole state set, then $w$ is said to be a *synchronizing word* (other commonly used name is *reset word*) and the automaton is *synchronizable.*

To put it clear, an automaton $\mathcal{A} = \langle Q, \Sigma \rangle$ is synchronizable if there is a word $w \in \Sigma^*$ such that $|Q \cdot w| = 1$. The previous definition is the most common found in the literature. In other terms a synchronizing word sends every state to an unique state. No matter in which state the computation begins it will always end in the same place.

One of the proposers of the modern concept of automaton was Edward F. Moore [29] in 1956. The automata proposed by Moore are essentially the same as the ones described here, the main difference was that the automata would return an output depending on the state they are after receiving each input letter. He proposed this concept as a way to make *"thought experiments"*(or Gedankenexperiment); to make an experiment is, precisely, to input a word on an automaton and take note of the output that comes out. Moore also conceived the automata as *black boxes* where just the inputs and outputs are known. A natural problem from this concept is to know if its possible to create an automaton and an input such that after the execution of the experiment one can determine with certainty the final state the automaton is in depending on the output received. This without regard of the state the experiment began. It was from this that Jan Černý took his inspiration to propose synchronizable machines in [13] (translated to English in [14]) in 1964. Černý considered the case when there is no output words received from the automata. But this was not the first time the notion was proposed. Already in 1963, Chung Laung Liu in his dissertation [27], presented to the MIT, devoted a whole section to, as he called them, "synchronizable" automata; hence one of the origins of the term. In contrast Černý used the term "directable".

There are two main reasons why the notion of synchronizable, or directable, automata was proposed several times in different places. The first one was the unavoidable lack of communication present in the pre-internet era. Černý's article was published in Slovak and presented in Europe; on the other hand Liu's work was made in USA and not published by that time. Other reason is the wide usefulness and naturalness of synchronization (some examples of this will be briefly discussed next) making it appear in several contexts under different appearances.

As a common motivation for this synchronization concept, Černý and Liu considered the task of recovering the control of an automaton. This means, that an user can determine without doubt the state in which the automaton is after introducing a series of inputs; this without consideration the state the automaton is the moment. This property is useful in situations where the communication between the automaton and the user is not always assured. Imagine, for example, a satellite that travels over the dark side of the moon, where it is impossible to receive or send any signal. Once the operators gain control over the satellite they do not know in which state the machine is.

The input of a reset word would help to retake the control over the satellite without problem. Another task that motivates synchronization, mentioned in Liu's work, is to control several copies of the same automaton that are in different states. It can be useful the existence of an input that makes them all to be in the same state, thus *synchronizing* them.

As a third motivation Liu mentions the interchange of codified messages, where a possible error could make the receiver to decodify differently the message (again for some lost connection) and the possibility of recover the message with a synchronizing input. A codification is the replacement of characters from an alphabet for words of (possibly) a different alphabet. This is a very natural thing to do nowadays as all digital information has to be replaced by strings of bits. A challenge comes at the time of decoding, going from the replacing alphabet to the original one. The difficulty appears if some characters are encoded in words of different length, in order to economize resources at the moment of transmission. There is a possibility that some encodings are prefixes of others making the communication prone to mistakes. This can be avoided creating a *prefix code*, a code such that no word is a prefix of another in the set of replacements. To decodify a message in a prefix code, an automaton can be used, the *decoder automaton*.

If in the coding there is a word $z$ such that when prefixed by any word $y$, both words in the same code alphabet, the resultant word $yz$ can be decomposed in coding words, then $z$ is called a synchronizing word and the code *synchronized*. It is not surprising that the decoder automaton of a synchronized code is synchronizable itself. And this was one of the motivations that Liu mentioned in his work. For a whole in depth discussion about codes and their automata we recommend to see [7].

Now, for something completely different, imagine a conveyor belt that transports industrial pieces. These pieces are all equally shaped but they can come in different orientations (for simplicity we are going to assume the number of possible orientations is limited). Furthermore imagine that it is possible to put some well placed obstacles that change the orientation to determined positions; some tall and some low obstacles that reorient the pieces. This behaviour can be modelled as an automaton, where the states are the positions of the piece and the letters represent the obstacles highness. Then it would be very convenient said automaton to be synchronizable in order to have a single sequence of obstacles that let all the passing pieces to a single determined orientation. A more detailed discussion of this example is present in [1]. See [30] and [31] for an example of the apparition of the use of synchronization without the explicit mention of automata.

These are some of the motivations that inspire the research in synchronizable automata. As it is evident the notion is versatile and very useful in different fields.

Once defined the concept of synchronization the next natural question that arises is: given an arbitrary automaton $\mathcal{A}$, how to decide if it has a reset word or, what is the same, if it is synchronizable? As it happens in many situations there is a conceptually easy answer for this that fails once the time to put it in practice comes. Let us see: let $\mathcal{A} = \langle Q, \Sigma \rangle$ be an arbitrary automaton. Denote the set of all non-empty subsets of $Q$ by $\mathcal{P}(Q)$. Note that every letter sends each non-empty subset to a unique non-empty subset, thus we can define a new automaton $\mathcal{P}(\mathcal{A}) := \langle \mathcal{P}(Q), \Sigma \rangle$, with the same alphabet and where its states are the non-empty subsets of the state set of the original automaton. This new automaton is called the *power automaton* of $\mathcal{A}$. Once constructed the power automaton from an arbitrary $\mathcal{A}$, to decide if there is a reset word for $\mathcal{A}$ it is just necessary to find a path from the state $Q$ to any state that represents a singleton, $\{q\} \subset Q$. This can be easily done constructing the graph of $\mathcal{P}(\mathcal{A})$ and using a Depth First Search algorithm to determine if such a path exists. In theory the previous procedure is fairly simple but its implementation is, to a certain point, impractical. This is due to the exponential growth of states of the power automaton with respect to the number of states of the original. If the state set of $\mathcal{A}$ has size $n$, then its power automata $\mathcal{P}(\mathcal{A})$ will have $2^n - 1$ different states.

Another point of intersection between Liu's [27] and Černý's [13] foundational works is a more convenient characterization of synchronizable automata. An automaton $\mathcal{A} = \langle Q, \Sigma \rangle$ has a reset word if and only if there is a word that synchronizes every subset of size 2, $\{p, q\} \subset Q$. This is, if for every pair of different states there is a word that sends them to the same state. This suggests a more effective algorithm to determine if any given automaton is synchronizable. Given an automaton $\mathcal{A} = \langle Q, \Sigma \rangle$, construct the graph of the power automaton considering only the subsets of size 2 and 1, and for every pair decide if there is a path that connects it with a singleton. If $|Q| = n$ then the amount of pairs is $\frac{n(n-1)}{2}$, therefore the previously suggested construction requires $O(|Q|^2 \cdot |\Sigma|)$ time, i.e. polynomial in the size of the original automaton, a much more reasonable time than its exponential alternative.

## The length of reset words

Once decided that an automaton has at least one synchronizing word, it is convenient to know how long these synchronizing words could be. Our interest is put in looking for the short words. More specifically how long can be the shortest synchronizing word of a given synchronizable automaton. Recall that the length of a word is the number of letters, not necessarily different, composing the word, e.g., the word *abbbaab* is of length 7. For a synchronizable automaton $\mathcal{A}$, denote $\mathrm{rt}(\mathcal{A})$, the *reset threshold* of $\mathcal{A}$, as the length of its shortest reset words. Once again, at the same time of presenting the concept of synchronization, Černý proposed the following series of automata $(\mathcal{C}_n)_{n\geq 2}$ with $\mathcal{C}_n = (\{1, 2, \ldots, n\}, \{a, b\})$ where $i \cdot a = i + 1$ for $1 \leq i \leq n-1$ and $n \cdot a = 1$; $1 \cdot b = 2$ and $i \cdot b = i$ for $2 \leq i \leq n$. See Figure 3.

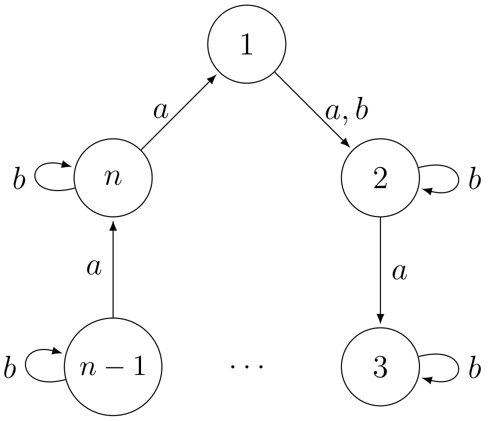


Figure 3: The automaton $\mathcal{C}_n$

Then he showed that each $\mathcal{C}_n$ is synchronizable and its shortest reset word is of the form $(ab^{n-1})^{n-2}a$, thus proving that $\mathrm{rt}(\mathcal{C}_n) = (n-1)^2$.

For each natural $n \geq 2$ denote $\mathfrak{C}(n)$ as the maximal reset threshold of all the synchronizable automata with $n$ states. In other words, given any reset automaton with $n$ states we can assure that the shortest reset word of this automaton is not longer than $\mathfrak{C}(n)$. Černý with his series of automata proved that:

$$(n-1)^2 \leq \mathfrak{C}(n) \leq 2^n - n - 1.$$

The upper bound is trivial since it is the number of non-empty subsets of size bigger than one. Few years later Peter Starke [36] (translated to English in [37]) improved the upper bound down to $1 + \frac{n(n-1)(n-2)}{2}$ and hypothesized for first time that $\mathfrak{C}(n) = (n-1)^2$. As is it usual in mathematics this hypothesis was misnamed as *Černý's Conjecture*. This conjecture has been open since then. The history of this conjecture has showed a lot of interesting and deep results. We recommend the very complete survey made by Mikhail V. Volkov in [38] to understand its history.

## Completely reachable automata

Although Černý's Conjecture is still open at the time writing for the general case, it has been proven for several sub-classes of automata. It is worth to highlight the case of *circular automata*, where one letter acts as a cyclical permutation. Louis Dubuc in [16] proved the conjecture for this kind of automata. This, together with other several examples, suggests it is worthwhile to focus on some particular classes of synchronizable automata.

Given an automaton $\mathcal{A} = \langle Q, \Sigma \rangle$, we consider a subset of states $P \subset Q$ to be *reachable* if there is a word $w \in \Sigma^*$ such that $Q \cdot w = P$. If we consider words as functions from the state set to itself, a subset is reachable if it is the image of some word. We say an automaton $\mathcal{A}$ is *completely reachable* if every non-empty subset is reachable. Note that an automaton is synchronizable if some subset of size one is reachable. That is why complete reachability is a specialization of the concept of synchronization.

The study of complete reachable automata began, not explicitely, from the studies of language complexity by Marina Maslennikova [28]. In this work she studied the state complexity of the languages of synchronizing words of certain automata. The state complexity of a regular language is the minimum amount of states that an automaton must have to recognize the language. Maslennikova proved that the complexity of the language of all reset words of Černý's automata grows exponentially with the size of the automata. To be more precise: She proved that to recognize all the reset words of $\mathcal{C}_n$, one needs an automaton of no less than $2^n - n$ states. The minimal automaton that she constructes for this task is the power automaton $\mathcal{P}(\mathcal{C}_n)$, with all the singleton subsets collapsed into one. Without explicitly noticing, Maslennikova proved that Černý's automata, among others, are completely reachable. Although it is not the only reason, the fact that the class of completely reachable contains Černý's series of automata motivated a further study on automata with this property.

Bondar and Volkov took the baton in [9] to continue the research of completely reachable automata. In this initial work, Bondar and Volkov properly define the concept of complete reachability and embark on the task of studying these new kind of automata. For this, they pay attention to two key aspects of a word: its excluded and duplicated states. Consider words as transformations of the set of states. If the size of the image of a word is smaller than the size of the whole set of states, this means that there are states which do not have preimage by this word, and, moreover, that there are states who have more than one state as preimage. For a given word $w$ (of a fixed but arbitrary automaton), the former set of states, without preimage, is called the *excluded set* of this word; and the latter set of states, with multiple preimages, is called the *duplicate set*. The size of the excluded set of a word is also relevant and it is called the *defect* of the word. From these terms Bondar and Volkov suggest the construction of a graph that connects the excluded states with the duplicate ones; this, at first, for words of defect 1. The connectivity of this graph was a sufficient but not necessary condition for automata being completely reachable.

Although the formulation and use of these graphs to prove complete reachability find their origin in [9], its inspiration can be traced back to Igor Rystsov's work, more exactly to his paper [33]. There he considers automata with two type of transformations: permutations and, what he calls, simple idempotents. In our terminology, simple idempotents are transformations of defect 1 such that every state is sent to itself, except for the excluded state that is sent to the duplicate state. Automata of this kind are called SI-automata. In his paper Rystsov proves that if an SI-automaton is strongly connected, then the automaton is sychronizable and, if there are $n > 1$ states, there is reset a word of length at most $2(n-1)^2$. For this, he uses a graph that connects all the excluded to the duplicate states of all the idempotent letters. Then, he *rotates* these edges using permutations of, each time, bigger lengths until a strongly connected graph is obtained. Rystsov proves that permutations of length no more than $2(n-1)$ are needed. For this we will call the graphs defined in [9] and [10] to characterize complete reachability *Rystsov graphs*.

In [10], Bondar and Volkov successfully found a characterization of completely reachable automata. For this, they extended the construction of the previously mentioned graph to account for words of bigger defect. They proposed a recursive construction of finite series of graphs. Each round has the strongly connected components of the previously constructed graph as its vertex set and uses words of bigger defect to connect these vertices. This construction finishes after a number of iterations bounded by the number of states of the automata. Using that construction, Bondar and Volkov proved that a given automaton is completely reachable if and only if the graph, constructed after the whole process, is strongly connected. The main argument of the proof of this is to show that every non-empty subset of states is expansible, i.e., the preimage by a word is bigger than the set. A revision and enhanced version of this process and proof appears in [8].

We come back to this particular construction in the following chapter since it is a core part of this work. There a detailed explanation is given.

From these two works, the study of completely reachable automata as a defined kind of automata has started.

Before we continue, it is convenient to highlight the work done by Henk Don in [15]. In this paper Don considers automata $\langle Q, \Sigma \rangle$ such that for every state $q \in Q$ there is a word of defect 1 that excludes $q$. He calls these automata *1-contracting*. Using a different language, Don proves that if a 1-contracting automaton $\mathcal{A}$ produces a Rystsov graph with a complete cycle, then $\mathcal{A}$ not only is synchronizable but completely reachable; and every non-empty subset of states is reachable by a concatenation of words of defect 1. Additionally, Don proves that if the 1-contracting automaton $\mathcal{A}$ has $n > 1$ states and for every state $q$ there is a word of defect 1 of length at most $n$ that excludes $q$, then not only $\mathcal{A}$ fulfils Černý's conjecture but every subset of size $1 < k < n$ is reachable by a word of length at most $n(n-k)$.

At the end of this paper Don proposed the conjecture that for an arbitrary automaton with $n > 1$ states if a subset of size $1 < k < n$ is reachable, then there is a word of length at most $n(n-k)$ that reaches it. This ended to be a very strong conjecture that would, among others, deduce Černý's conjecture. Effectively this conjecture was proven false by Gonze and Jungers in [21]. In this work they showed two series of automata: the first one where for automata of size $n$ congruent with 3 modulo 4, there are sets of size $n-2$ reachable with words no shorter than $O(n^2)$; the second series of automata augments the gap, with subsets of size $\lfloor \frac{n}{2} \rfloor - 1$ (for $n > 6$) such that the shortest words that reach them have length of $2^n/n$. Not only these counter examples were the contribution of this paper. It is worth to mention the propose of a quadratic time, on the number of states, algorithm that allows to calculate the Rystsov graph proposed in [9]; and with this disproved some conjectures proposed in the same paper.

Deciding if a given automaton is completely reachable is the first natural problem to solve once the concept has been introduced. The characterization presented [10] and [8], by the moment, is of little practical help since it is not known what is the complexity of obtaining all the words of certain defect, or at least the needed ones to construct the graphs. A great advance in the determination of the complexity of deciding complete reachability was made by Ferens and Szykuła in [17] who proposed an algorithm that decides in polynomial time, on the number of states, whether a given automaton is completely reachable or not. The idea of the algorithm is to find the largest subset that is not expandable. Both conditions of being the largest and non-expandable imply that this subset is not reachable. If the proposed algorithm does not return any subset, it can be concluded that the automaton given as input is completely reachable. Following this, Ferens and Szykuła propose another algorithm to find a short reaching word for any subset of states of an automaton, given the case, of course, that this subset is reachable. Using this algorithm it is obtained a partial but important result in regards to the length of reaching words. Although Ferens and Szykuła do not prove Don's conjecture for completely reachable automata, they prove that if the automaton $\mathcal{A}$ has $n > 1$ states and is completely reachable, then for every subset of size $1 < k < n$ there is a word that reaches this subset with length no bigger than $2n(n-k)$.

There are other considerations about completely reachable automata that go beyond the algorithmic and the bounding of the reaching words. It is worth mentioning the works of Stefan Hoffmann in [24] and [23]. In [24] Hoffmann characterizes primitive groups of degree[1] at least 5 as such permutation groups that in the presence of any transformation of defect 1 the resultant automaton is completely reachable. Moreover he relates the capacity of a permutation group of degree $n > 3$ to connect every subset of size $1 < k < n$ (being $k$-homogeneous) to the behaviour of this group together with a transformation of rank $k$. In [23], Hoffmann retakes the subject of the complexity of the language of synchronization words. There he proves a sufficient condition for binary automata such that their synchronization language has maximum complexity; then he proceeds to add examples, to the already given in [28], of series of automata with maximal synchronization complexity.

As we have seen, the study of completely reachable automata is a relatively new edge of an old problem. Nevertheless it has proven to be a fertile field for new and interesting discoveries and problems. In this work we see some of these new discoveries.

# Goals and objectives of the research

The main goal of the work is to further the study of completely reachable automata. For this we traced the following more specific objectives:

- To find alternative and reliable characteristics and methods to determine whether an automaton is completely reachable.

- Next, is to study the length of the words reaching each non-empty subset.

- More precisely, to determine whether or not Don's conjecture is satisfied by some kind of automata.

In a transformation from a finite set to itself, i.e., $f\colon Q \to Q$, we can consider two sides: the *image* and the *kernel*. The kernel is a partition of $Q$ joining in the same subset states with the same image by $f$. An additional

[1] The amount of points the group acts on. In this case the number of states of the automaton.

objective is to study automata where for any possible partition of the set of states there is a transformation such that its the kernel is the partition.

## Scientific novelty of the research

Besides some results stated to contextualize the discussion, for which we give the respective credit to who deserves it, every result is a new contribution to the literature of the subject treated in the dissertation.

## Theoretical and practical significance of the research

The dissertation work is theoretical in its nature. The results obtained push further the knowledge not only about the main subject which is completely reachable automata, but also about its original subject that is synchronization of automata. From this, some of the algorithms described run in polynomial time, what it is always a desired characteristic in this area. This lays a stable background for any practical implementation needed by further research.

## Methodology of research

The research conducted was mostly theoretical. It used knowledge from several fields of mathematics and theoretical computer science. Among then we can highlight theory of automata, algorithms, graphs and finite groups.

## Overview of the thesis and our contributions

In Chapter 1 we establish the necessary definitions and results to give context to this work. The main part of this section is the explanation of the construction of Rystsov graphs given in [8]. Since a great part of the work done here is based on these graphs, to show their construction process helps to give a better context to the following work. Additionally to the necessary definitions and statements, we present an algorithm to calculate the Rystsov graph of an arbitrary automaton. The construction of the Rystsov graph of any automaton requires the iterative construction of intermediate graphs. In principle these intermediate steps are not easy to compute in terms of complexity. But we present an scheme to calculate all the necessary components of this intermediate steps. This derive in Theorem 1.5 that tell us we can construct this intermediate steps in a quasi polynomial time.

In Chapter 2, we focus our attention on binary automata, i.e., those with only two letters. There we prove that for automata with more than two states in order to them being completely reachable, one of the letters must be a cyclic permutation over all the states and the other must have defect 1. But this is not enough, besides that, the letter of defect 1 must not preserve subsets of states that represent subgroups of the cyclic group of the same size of the automaton. Proposition 2.1 states that a binary completely reachable automaton does not preserve the aforementioned subsets. After that, Proposition 2.5 says that the invariance of these *subgroups* ensures complete reachability; while Proposition 2.6 shows what happens when this condition is not met. The combination of these propositions converge in Theorem 2.1 that gives a characterization for binary completely reachable automata. This derives in an almost linear time algorithm to decide complete reachability for these kind of automata.

We continue the discussion of binary completely reachable automata in Chapter 3. In this chapter we consider the length of the words reaching the subsets. Our main result is Theorem 3.1. To prove it we consider the expansion method. A word properly expands a subset if the word does not have excluded elements in the subset and the preimage by this words is bigger. In a completely reachable automaton every subset has a word that properly expands said subset. The expansion method aims to bound the length of words that properly expand every subset. With Proposition 3.3 we prove a linear bound (with respect to the size of the automaton and the subset) for words that expand subsets of binary completely reachable automata with a particular condition. Additionally it is shown why this method can not be used for the general case of binary automata.

For Chapter 4 we extend the results of Chapter 2. We consider automata with just one letter of defect 1 but where the rest of the letters are permutations of the state set. These are called *almost group automata.* We add the condition that the group generated by the permutation letters is not only transitive but imprimitive, since in other case (if the group is primitive) it is already known the automaton is completely reachable.

In this case complete reachability depends on the letter of defect 1 not preserving the blocks of imprimitivity that contain its excluded state. Similarly to the case of binary automata Proposition 4.1 proves the necessity of this condition for complete reachability. After that we describe the Rystsov graphs of these kind of automata; this is done in Proposition 4.2. In comparison with the case of binary completely reachable automata, we need an additional condition to prove that not preserving blocks is sufficient. This is stated in Theorem 4.2.

In Chapter 5 we shift our attention from reachable subsets to partitions of the state set. Every transformation not only defines a subset of states, the image, but a partition or equivalence relation on the states set. Roughly speaking two different states are related by a transformation if they have the same image. Through all the this work we have focused our attention on the subset, image, part of transformations and the possibility obtaining them all. In this chapter we aim to study automata that can realize every possible partition with a transformation. We name these automata *totally compatible* and prove a characterization for them. This characterization is stated in Theorem 5.1. Again, transformations of defect 1 are the key for characterizing totally compatible automata. Then we proceed to describe a simple algorithm with polynomial time complexity to recognize them. At the end we prove that there is a connection between some kind of these automata and completely reachable ones.

## Publications, seminars and conferences

The main results of this work were published in the following papers:

- Eugenija A. Bondar, David Casas, and Mikhail V. Volkov. Completely reachable automata: An interplay between automata, graphs, and trees. *International Journal of Foundations of Computer Science*, 34(06):655– 690, July 2023.

- David Casas and Mikhail V. Volkov. Binary completely reachable automata. *In LATIN 2022: Theoretical Informatics*, pages 345–358. Springer International Publishing, 2022.

- David Casas. A Characterization of Totally Compatible Automata. *Journal of Automata, Languages and Combinatorics*, 27(4), pages 249-257, 2022.

- David Casas and Mikhail V. Volkov. Don's conjecture for binary completely reachable automata: an approach and its limitations. To appear in *Journal of Automata, Languages and Combinatorics*, 29(2-4) 2024. A preprint can be found in: *https://arxiv.org/abs/2311.00077.*

- David Casas. Completely reachable almost group automata. *Ural Mathematical Journal*, Vol. 10, no.2, pp.37-48, 2024.

Most of the work presented in this dissertation was made together with Mikhail V. Volkov. The joint contributions are indistinguishable in the majority of this dissertation. But the author would like to be more specific in the case of the work presented in Chapter 1. This definitions and results presented in this chapter are developed with more detail in [8]. In this case it is due to attribute the theoretic work to Eugenija A. Bondar and Mikhail V. Volkov, and the discussion presented in Subsection 1.2 to the author.

Additionally, some of the results of this work were reported in the following events:

- Seminars of Algebraic Systems. Institute of Natural Sciences and Mathematics. Ural Federal University. Yekaterinburg, Russia.
- International (52-th) Youth School-Conference of Modern problems in mathematics and its applications, Yekaterinburg, Russia, 2021.
- LATIN 2022: The 15th Latin American Theoretical Informatics Symposium, Guanajuato, Mexico, 2022.

# Acknowledgments

There is no sizable work done due to the efforts of a single person, despite what the title page says. I want to thank Professor Mikhail V. Volkov for his guidance, patience and teachings. His remarks in several aspects of this work made it much better than it could be; his wise suggestions and teachings helped the results here presented to see the light; and his, overall, generosity allowed this endeavor to bear fruit. I would also like to thank the different people that I met here and made this experience more pleasant or interesting.

Также я хотел бы поблагодарить Надежду Николаевну. Ее добросовестная работа и усердие очень помогли в нескольких, возможно, недооцененных, но важных случаях.

Le doy mil gracias a la gente que desde lejos me acompañó y apoyó con sus palabras de ánimo. Le agradezco a mi familia padre, hermano y especialmente a mi madre, porque aportaron, entre otras muchas cosas, la tranquilidad y confianza en mí mismo que tantas veces faltaron durante este proceso. A los amigos que con su alegría y compañía hicieron más calidas las frías jornadas. Especialmente a Jenny por su generosidad y siempre oportunas palabras. A Angélica que estuvo presente en varios momentos, brindó su desinteresada ayuda en ocasiones difíciles y sobre todo hizo mi vida un poco más interesante. A todas las diferentes personas que creyeron en mí y ayudaron en diferentes formas que ni ellos mismos son conscientes.

Finally, I want to thank UrFU and its staff for the good work and welcoming assistance they have shown these years. I also thank the Ministry of Science and Higher Education of the Russian Federation for the grant that allowed me to initiate and finish my studies.

# Chapter 1

# Basics

## 1.1 Automata and graphs

In this section we give the formal definitions and results needed to understand this work. First let us introduce our main protagonists:

**Definition 1.1.** A *Deterministic Finite Automaton* (DFA), or simply an *automaton*, is a triple $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$, where:

- $Q$ is a finite set, called the set of *states*,
- $\Sigma$ is a finite set of letters. The *alphabet*,
- and $\delta \colon Q \to \Sigma$, is the transition function.

For each letter of the automaton's alphabet $a \in \Sigma$ we can define the function $\delta_a : Q \to Q$ where $\delta_a(q) := \delta(q, a)$, whence each letter can be considered individually as a function of $Q$ to itself or a *transformation* over $Q$. This allows us to not consider the whole transition function but the functions over $Q$ defined by each of the letters. Thus we can use the following notation: for every $q \in Q$ and $a \in \Sigma$ we will denote $\delta(q, a) := q \cdot a$. According to this, an automaton can be specified just with its set of states and the action of each letter in this set; that is why from now on we will define automata as pairs of the state set and the alphabet, i.e., we write $\mathcal{A} = \langle Q, \Sigma \rangle$.

Before we proceed with further definitions related with automata, let us introduce some vocabulary of graph theory and some notation related to it. A *directed graph* $\Gamma$ is a pair $(V, E)$, where $V$ is a finite set the *vertex set* and $E \subseteq V \times V$ is the set of *directed edges*. Since in this dissertation we consider only directed graphs, from now we omit the word "directed". For reference, the first and second components of an edge are called the *source* and *target* respectively. More often than not we consider *labelled* graphs. A *labelling* of a graph $\Gamma = (V, E)$ is a function $\lambda : L \to E$ where $L$ is a finite set. We denote an edge $(s, t) \in E$ labelled with $w \in L$ as $s \xrightarrow{w} t$. This definition emulates the existence of "parallel" edges, in the sense of one edge with multiple labels represents different edges with the same source and target. A *path* of a graph is a set of edges $e_1, e_2, \ldots, e_m$, with $m \geq 1$, such that for every $1 \leq i < m$, the target of $e_i$ is the same as the source of $e_{i+1}$. Vertices $p, q \in V(\Gamma)$ are *strongly connected* if there is a path from $p$ to $q$ and from $q$ to $p$. We consider each vertex as strongly connected with itself. A *strongly connected component* of a graph is a maximal subset of vertices such that every vertex is strongly connected to each other. At the same time a graph is called *strongly connected* if there is only one strongly connected component,i.e., the vertex set.

An automaton $\langle Q, \Sigma \rangle$ can be represented as a labelled graph, where the vertex set is $Q$ and for each state $p \in Q$ and letter $a \in \Sigma$, there is a labelled edge $p \xrightarrow{a} p \cdot a$. This is the *underlying* graph of the automaton.

To a better understanding of the previous definitions and further discussion a graphical representation is of great help. With this in mind consider the following automaton: $\mathcal{C}_4 = (\{1, 2, 3, 4\}, \{a, b\})$, the action of the letters are defined in the following way:

| $q$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $q \cdot a$ | 1 | 2 | 3 | 1 |

,

| $q$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $q \cdot b$ | 2 | 3 | 4 | 1 |

.

The underlying graph of this automaton is represented in the Figure 1.1.

The elements of $\Sigma$ are called *input letters* and finite sequences of letters are called *words over* $\Sigma$. The empty sequence, without any letter, is also treated as a word, called the *empty word* and denoted by $\varepsilon$. Let $\Sigma^*$ be the set of all words of the alphabet $\Sigma$. The action of a word $w \in \Sigma^*$ over every state $q \in Q$ can be defined recursively: $q \cdot \varepsilon := q$; and if $w = va$ with $v \in \Sigma^*$ and $a \in \Sigma$, then $q \cdot w := (q \cdot v) \cdot a$. Note that any word $w \in \Sigma^*$ produces a transformation over the set of states, this is, the consecutive composition of the transformations defined by each letter of $w$.

The set $T(\mathcal{A})$ of all transformations induced this way is called the *transition monoid* of $\mathcal{A}$; this is the submonoid generated by the transformations $q \mapsto q \cdot a$, with $a \in \Sigma$, in the monoid of all transformations of $Q$. An automaton $\mathcal{B} = \langle Q, \Theta \rangle$ with the same state set as $\mathcal{A}$ is said to be *syntactically equivalent* to $\mathcal{A}$ if $T(\mathcal{B}) = T(\mathcal{A})$.

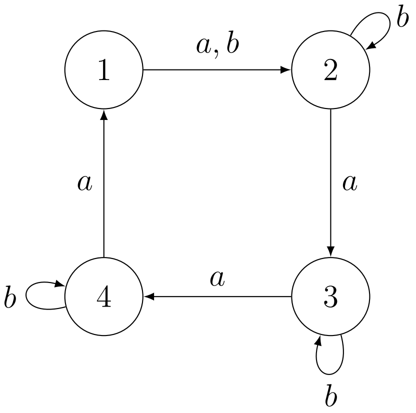


Figure 1.1: Example of the graph of the automaton $\mathcal{C}_4$

The action of words can be applied to subsets of states: if $P \subseteq Q$ and $w \in \Sigma^*$, then $P \cdot w := \{p \cdot w \mid \text{ for every } p \in P\}$.

Given an automaton $\mathcal{A} = \langle Q, \Sigma \rangle$ and a word $w \in \Sigma^*$, the *image* of $w$ is the set $Q \cdot w$ and the *excluded set* of $w$, denoted by $\mathrm{excl}(w)$, is the complement of the image, i.e., $Q \backslash Q \cdot w$. The number $|\mathrm{excl}(w)|$ is called the *defect* of $w$. If a word $w$ has defect 1, its excluded set consists of a unique state called the *excluded state* for $w$. Further, for any $w \in \Sigma^*$, the set

$$\{p \in Q \mid p = q_1 \cdot w = q_2 \cdot w \text{ for some } q_1 \neq q_2\}$$

is called the *duplicate set* of $w$ and is denoted by $\mathrm{dupl}(w)$. If $w$ has defect 1, its duplicate set consists of an unique state called the *duplicate state* for $w$. We identify singleton sets with their elements, and therefore, for a word $w$ of defect 1, $\mathrm{excl}(w)$ and $\mathrm{dupl}(w)$ stand for its excluded and, resp., duplicate states.

For any $v \in \Sigma^*$, $q \in Q$, let $q \cdot v^{-1} := \{p \in Q \mid p \cdot v = q\}$. Then for all $u, v \in \Sigma^*$,

$$\mathrm{excl}(uv) = \{q \in Q \mid q \cdot v^{-1} \subseteq \mathrm{excl}(u)\}, \tag{1.1}$$

$$\mathrm{dupl}(uv) = \{q \in Q \mid q \cdot v^{-1} \cap \mathrm{dupl}(u) \neq \varnothing \text{ or } |q \cdot v^{-1} \backslash \mathrm{excl}(u)| \geq 2\}. \tag{1.2}$$

The equalities (1.1) and (1.2) become clear as soon as the definitions of excl( ) and dupl( ) are deciphered. Figure 1.2 illustrates the meaning of these equalities.

Given a transformation or, which is the same, a word $w$ over $\Sigma$, consider the following relation:

$$\ker(w) := \{(p, q) \in Q \times Q \mid p \cdot w = q \cdot w\}.$$

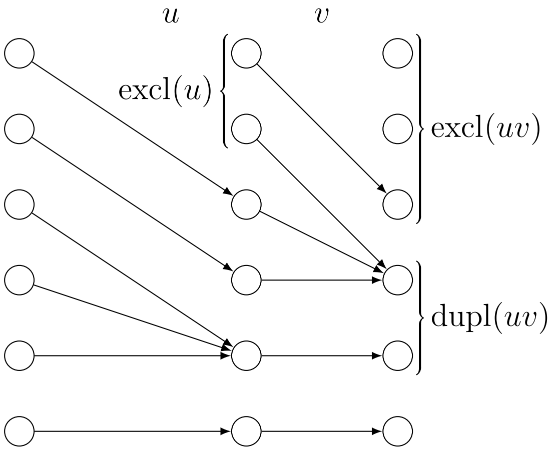


Figure 1.2: An illustration of equalities (1.1) and (1.2)

It is easy to see that this is an equivalence relation, which is called the *kernel* of the transformation $w$. The equivalence relation that produces an unique class (all the states are related) and the one that produces one class for each state (each element is related just with itself) are called the trivial equivalence relations. Additionally, for the case of words of defect 1 we know that exactly two states must have the same image; we will call this pair of states the *collapsed set* of the word, denoted by coll().

Some transformations over a set of states $Q$ can be bijective, or what is called *permutations*. Thanks to this we can use some terminology of the theory of permutation groups. Recall that the set of all the bijective transformations of a finite set $Q$ to itself is denoted by $S_Q$, also called the *symmetric group* of $Q$. Let $G \subset S_Q$ be a group of permutations. This group is said to be *transitive* if for every pair of states $q, p \in Q$ there is a permutation $g \in G$ such that $p \cdot g = q$. A non-empty subset $B \subseteq Q$ is said to be a *block* of the group if and only if for every $g \in G$ either $B \cdot g = B$ or $B \cdot g \cap B = \emptyset$. The singletons and $Q$ itself are, always, blocks, these are called *trivial.* A permutation group $G \subseteq S_Q$ is said to be *primitive* if it is transitive and the only blocks are the trivial ones; otherwise the group is said to be *imprimitive.* When we talk about a block of imprimitivity in the present work, unless stated the contrary, it will always be non-trivial. If a transitive group $G \subset S_Q$ has a block of imprimitivity $B \subseteq Q$, the images of $B$ by $G$ are also blocks of imprimitivity and form a partition of $Q$. This collection of sets, the partition, is called a *system of imprimitivity* of the group $G$.

The following definition is illustrative and useful in key parts of our work[1].

**Definition 1.2.** Let $G$ be an arbitrary group and $X$ be a subset of $G$. The (*right*) *Cayley digraph of* $G$ *with respect to* $X$ is a graph with $G$ as its vertex set and

$$\{(g, gx) \mid g \in G,\ x \in X\}$$

as its edge set. It is denoted by $\mathrm{Cay}(G, X)$,

Additionally to the definition the next property of Cayley digraphs of finite groups is folklore; this property is useful for the purposes of Chapters 2 and 3.

**Lemma 1.1.** *Let $G$ be a finite group, $X$ a subset of $G$, and $H$ the subgroup of $G$ generated by $X$. The strongly connected components of the Cayley digraph* $\mathrm{Cay}(G, X)$ *have the left cosets $gH$, $g \in G$, as their vertex sets, and each strongly connected component is isomorphic to* $\mathrm{Cay}(H, X)$.

Although synchronization of automata is not the main concern of this work, it is, indeed, the context from which all this work comes from; therefore it is useful to have at hand some of the formal definitions from this subject.

**Definition 1.3.** Let $\mathcal{A} = \langle Q, \Sigma \rangle$ be an automaton. If there is a word $w \in \Sigma^*$ such that for every pair of states $p, q \in Q$ it happens $p \cdot w = q \cdot w$, then it is said that the automaton $\mathcal{A}$ is **synchronizable** and that $w$ is a *synchronizing* or a *reset* word of $\mathcal{A}$.

**Definition 1.4.** Let $\mathcal{A} = \langle Q, \Sigma \rangle$ be an automaton and $\mathcal{P}(Q)$ the set of all the subsets of $Q$. Define $\mathcal{P}(\mathcal{A}) := \langle \mathcal{P}(Q), \Sigma \rangle$ the *power automaton* of $\mathcal{A}$, where for every $a \in \Sigma$ and $P \in \mathcal{P}(Q)$

$$P \cdot a := \{p \cdot a \mid a \in P\}.$$

## 1.2 Completely reachable automata and bounds of reachability

Let $\mathcal{A} = \langle Q, \Sigma \rangle$ be an automaton, a non-empty subset of states $P \subseteq Q$ is said to be *reachable* if there is a word (not necessarily unique) $w_P \in \Sigma^*$ such that its image is $P$, i.e., $Q \cdot w_P = P$. An automaton is called *completely reachable* if every non-empty subset of states is reachable. Note that if an

[1] In fact, our definition is the semigroup version of the notion of a Cayley digraph, but this makes no difference since in a finite group, every subsemigroup is a subgroup.

automaton is completely reachable, then it is synchronizable; but the other way around is not necessarily true.

Recall the series of automata proposed by Černy (see Subsection ); in [28] Maslennikova proved that every Černý automaton is completely reachable, adding another nice property to these series of automata.

It is natural to relate reachability with synchronization since the former is a specialization of the latter, hence statements about reachability can be used for synchronization, as we will see later.

Once defined the concept of complete reachability, the first question to comes to mind is: *given an automaton, how to decide whether or not it is completely reachable?* That is the subject of our next discussion.

### Rystsov graphs

Let $\mathcal{A} = \langle Q, \Sigma \rangle$ be an automaton. In order for $\mathcal{A}$ to be completely reachable, there must be words that reach the subsets of states of size one minus than the whole state set, i.e., words of defect 1. If $w \in \Sigma^*$ is a word of defect 1, recall the definitions of $\mathrm{excl}(w)$, as the single state that does not have pre-image; and of $\mathrm{dupl}(w)$ as the unique state with two pre-images. Note that for any word $w$ its defect does not decrease when letters are concatenated to the word. From this fact we can conclude that there must be at least one letter of defect 1 in $\Sigma$. For $\mathcal{A}$ define $\Gamma_1(\mathcal{A})$ as the graph with vertex set $Q$ and the edge set:

$$E\colon = \{(\mathrm{excl}(w), \mathrm{dupl}(w)) \mid w \text{ is a 1-defect word}\}.$$

For the edge $(\mathrm{excl}(w), \mathrm{dupl}(w)) \in E$ we say that the word $w$ *enforces* it. This graph is what Rystsov used in [33] as the main tool for his results. We call these together with the extensions presented in the next subsection *Rystsov graphs*. Recall that a graph is *strongly connected* (or simply *connected*) if for any pair of vertices there is a path from one to the other and way back.

These are the necessary definitions to state the following theorem:

**Theorem 1.1** ([9]). *Let $\mathcal{A} = \langle Q, \Sigma \rangle$ be an automaton, if $\Gamma_1(\mathcal{A})$ is strongly connected then $\mathcal{A}$ is completely reachable.*

The main idea of the proof is to work by induction, not in the size of the subsets but in size of their complements. Let $P \subseteq Q$ be a subset of states, the induction goes on $k = |Q| - |P|$. The base case, $k = 1$, comes from the fact that to $\Gamma_1(\mathcal{A})$ being strongly connected, thus every vertex, or state, must have an outgoing edge, what makes any set with one state absent reachable. After that, suppose the induction hypothesis for some $k \geq 1$, if $P$ is such that $|Q| - |P| = k + 1$, the condition of strong connectivity also allows to imply an edge going from outside to inside $P$. This makes possible to reach $P$ from a subset bigger by one, such that its difference has size exactly $k$, the induction hypothesis makes this bigger subset reachable and hence $P$ too.

It is natural to question if this condition is also necessary. This is not the case. Figure 1.3 shows the graph of the automaton $\mathcal{E}_3$, and the graph $\Gamma_1(\mathcal{E}_3)$. It can be directly calculated that $\mathcal{E}_3$ is completely reachable meanwhile $\Gamma_1(\mathcal{E}_3)$ is not strongly connected.

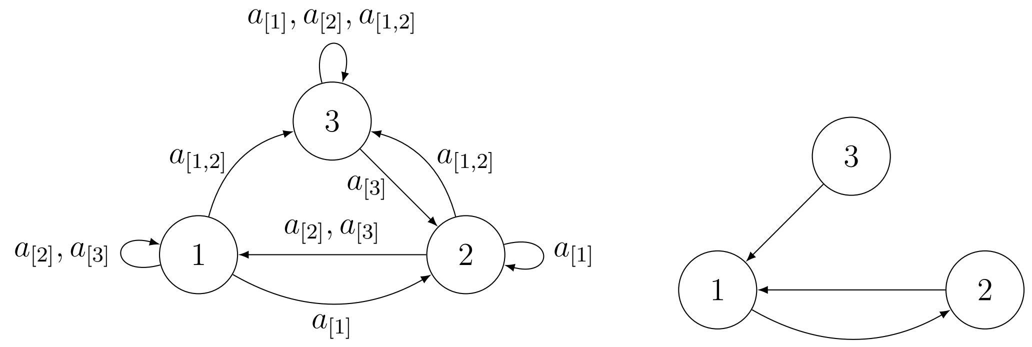


Figure 1.3: The automaton $\mathcal{E}_3$ and its graph $\Gamma_1(\mathcal{E}_3)$.

The proof of Theorem 1.1 suggests why the converse of the statement is false. Since all the edges represent words of defect 1, in reality what was proven in the theorem is that if the graph $\Gamma_1(\mathcal{A})$ is strongly connected every subset can be reached by a word made of certain number of subwords of defect 1; something that may not be always true. Nevertheless it would be natural to conjecture that if this is the case, any subset is reachable by a product of words of defect 1, then the graph of the automaton is strongly connected. This conjecture was proven false in [21]. The counter example is the automaton $\mathcal{P}_4$ with six states ($\{1, 2, 3, 4, 5, 6\}$) and six letters 1-defect ($\{a, b, c, d, e, f\}$) described in Table 1.1.

There it was proven that this automaton is completely reachable and its graph $\Gamma_1(\mathcal{P}_4)$ is not strongly connected.

With this, the question that is left is: *under which conditions is the graph $\Gamma_1(\mathcal{A})$ of an automaton strongly connected?* An automaton $\mathcal{A} = \langle Q, \Sigma \rangle$ is *perfectly reachable* if each subset with $k > 0$ states is reachable in $\mathcal{A}$ by a product of $|Q| - k$ words of defect 1. The following characterization of perfectly reachable automata, stated in [12], answers our question. The Rystsov graph $\Gamma_1(\mathcal{A})$ is strongly connected if and only if the automaton is perfectly reachable. The proof comes from a combination of two results in

|   | a | b | c | d | e | f |
|---|---|---|---|---|---|---|
| 1 | 2 | 4 | 1 | 1 | 1 | 1 |
| 2 | 3 | 6 | 2 | 2 | 2 | 5 |
| 3 | 4 | 5 | 5 | 3 | 3 | 3 |
| 4 | 5 | 1 | 6 | 6 | 4 | 4 |
| 5 | 6 | 3 | 4 | 5 | 6 | 2 |
| 6 | 2 | 3 | 4 | 5 | 6 | 2 |

Table 1.1: Tabular description of $\mathcal{P}_4$.

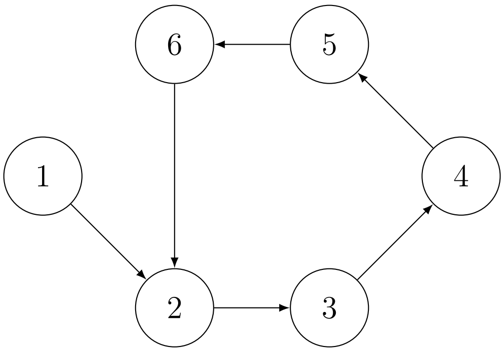


Figure 1.4: The graph $\Gamma_1(\mathcal{P}_4)$

the literature.

**Proposition 1.1.** *An automaton is perfectly reachable if and only if its Rystsov graph is strongly connected.*

*Proof.* The proof of [9, Theorem 1] shows that an automaton $\mathcal{A}$ is perfectly reachable whenever the graph $\Gamma_1(\mathcal{A})$ is strongly connected. The converse follows from [21, Theorem 20]. □

### A sufficient and necessary condition

The last section presented a construction, the Rystsov graph $\Gamma_1()$, together with a condition, strong connectivity, which together imply complete reachability. It is desirable to have a condition that unequivocally indicates that an automaton is complete reachable. This condition was presented in [10]. There the construction of the graph $\Gamma$ was extended recursively and from this extension a condition sufficient and necessary for complete reachability was proved.

Now let us proceed to describe the process to extend the construction of the graph. Given a automaton $\mathcal{A} = \langle Q, \Sigma \rangle$ first construct the graph $\Gamma_1(\mathcal{A})$.

Then test if this graph is strongly connected; in case it is, set $\Gamma(\mathcal{A}) = \Gamma_1(\mathcal{A})$ and finish with a SUCCESS. In other case verify the size of the strongly connected components. If all are singletons, then also set $\Gamma(\mathcal{A}) = \Gamma_1(\mathcal{A})$, but in this case finish with FAILURE. In other case continue with the process.

The construction of the total graph goes recursively depending on the previously constructed graphs. For $k \geq 1$ the vertices of the graph $\Gamma_{k+1}(\mathcal{A})$, if it makes sense to construct it, are the strongly connected components of $\Gamma_k(\mathcal{A})$. We denote the different vertex sets by $Q_k$. That is, for $k = 2$, the vertices in $Q_2$ are subsets of $Q$; and if we can continue, then the vertices of $Q_3$ are collections of subsets of $Q$ and so on. This forms a tree structure where a vertex $C$ in $\Gamma_k(\mathcal{A})$ is the child of a vertex $D$ in $\Gamma_{k+1}(\mathcal{A})$ if $C \in D$. From this observation we define the *foliage* of a vertex as follows: for every $C \in Q_2$ its foliage, denoted by $\operatorname{leaf}(C)$, is the set of states itself; and for $k \geq 2$, the foliage of $C \in Q_{k+1}$ is

$$\operatorname{leaf}(C) := \bigcup_{D \in C} \operatorname{leaf}(D).$$

From this recursive definition we can see that the foliage of a vertex in any level is a subset of states of the automaton.

Now let us continue with the recursive construction of the graph $\Gamma(\mathcal{A})$. Suppose that the graph $\Gamma_k(\mathcal{A})$ with $k > 1$ has been constructed. If this graph is strongly connected, then finish the process with a SUCCESS and set $\Gamma(\mathcal{A}) = \Gamma_k(\mathcal{A})$. In other case find its strongly connected components. These will be the vertices in $Q_{k+1}$. But before this we need to verify the size of each of the foliages of the new vertices. If all are smaller than $k+1$, then, also, set $\Gamma(\mathcal{A}) = \Gamma_k(\mathcal{A})$ and finish with FAILURE. If this is not the case, there is at least one strongly connected component with a foliage of size larger than $k+1$, we continue the construction of $\Gamma_{k+1}(\mathcal{A})$ with the vertex set $Q_{k+1}$. For $k \geq 1$ define $W_k(\mathcal{A})$ as the set of all words in $\Sigma^*$ with defect $k$. The edges will be those in $E_k$ connecting vertices in $Q_{k+1}$ together with the following set:

$$\begin{aligned} E_{k+1} := \{C \xrightarrow{w} D \in Q_{k+1} \times Q_{k+1} \mid C \neq D,\ & \text{there is a } w \in W_{k+1}(\mathcal{A}), \\ & \operatorname{excl}(w) \subseteq \operatorname{leaf}(C),\ \operatorname{dupl}(w) \cap \operatorname{leaf}(D) \neq \emptyset\}. \end{aligned}$$

In this case we extend the previous terms and say that the word $w$ *enforces* the edge $C \xrightarrow{w} D$. In this point an example is due. Consider the automaton

$\mathcal{E}_5$ with the set state $\{1, 2, 3, 4, 5\}$ and the following transition table:

| | $a_{[1]}$ | $a_{[2]}$ | $a_{[3]}$ | $a_{[4]}$ | $a_{[5]}$ | $a_{[1,2]}$ | $a_{[4,5]}$ | $a_{[1,3]}$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 1 | 1 | 1 | 1 | 3 | 1 | 4 |
| 2 | 2 | 1 | 1 | 2 | 2 | 3 | 1 | 4 |
| 3 | 3 | 3 | 2 | 3 | 3 | 3 | 2 | 4 |
| 4 | 4 | 4 | 4 | 5 | 4 | 4 | 3 | 5 |
| 5 | 5 | 4 | 5 | 5 | 4 | 5 | 3 | 5 |

Figure 1.5 shows the iterative construction of the graph $\Gamma(\mathcal{E}_5)$.

For an automaton $\mathcal{A}$ with $n$ states and a graph $\Gamma_k(\mathcal{A})$ not strongly connected the maximum size of any foliage is $n-1$ what implies that the process will stop, either in SUCCESS or FAILURE, in at most $n-1$ steps.

Now it is possible to state the two theorems that characterize completely reachable automata:

**Theorem 1.2** ([10]). *If an automaton $\mathcal{A} = \langle Q, \Sigma \rangle$ is such that the graph $\Gamma(\mathcal{A})$ is strongly connected and $\Gamma(\mathcal{A}) = \Gamma_k(\mathcal{A})$, then $\mathcal{A}$ is completely reachable; more precisely, for every non-empty subset $P \subseteq Q$, there is a product $w$ of words of defect at most $k$ such that $P = Q \cdot w$.*

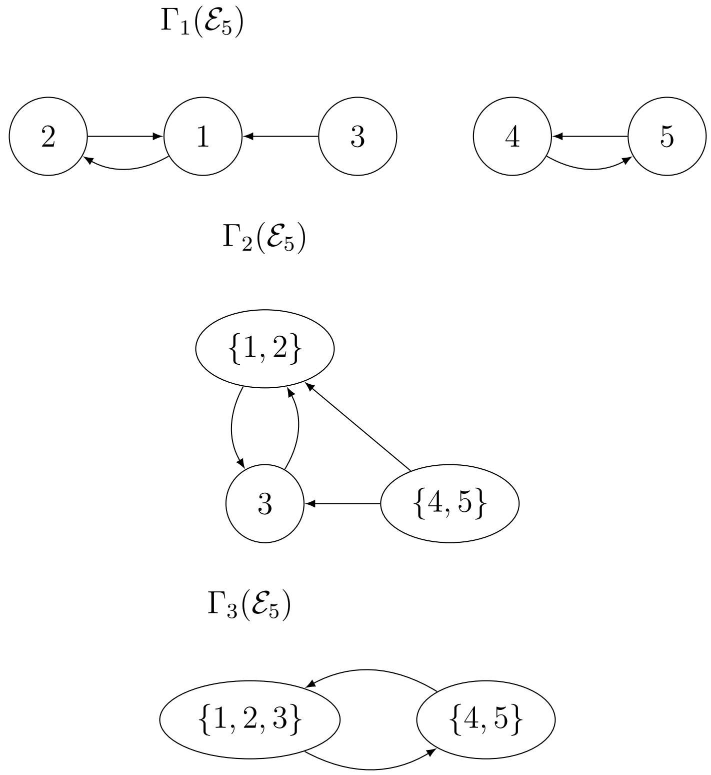
The proof of Theorem 1.2 uses the same idea that in Theorem 1.1: induction in the size of the difference $Q \setminus P$, for any non-empty subset $P \subseteq Q$. The difference here is that the useful edge does not necessarily come from an excluded state.

This theorem is an extension of the sufficient condition presented in Theorem 1.1. It tell us that if the process of constructing $\Gamma(\mathcal{A})$ ends with SUCCESS then the automaton is completely reachable. Now it is time to see what happens in the case of FAILURE.

**Theorem 1.3** ([10]). *If an automaton $\mathcal{A} = \langle Q, \Sigma \rangle$ is such that the graph $\Gamma(\mathcal{A})$ is not strongly connected, then $\mathcal{A}$ is not completely reachable; more precisely, if $\Gamma(\mathcal{A}) = \Gamma_k(\mathcal{A})$ and it is not strongly connected, then some subset in $Q$ with at least $|Q| - k$ states is not reachable in $\mathcal{A}$.*

If a graph fails to be strongly connected then in it there is at least one strongly connected component without entering edges from outside (a minimal strongly connected component). Using the hypothesis that this is the case for $\mathcal{A}$ let $D$ be the foliage of one minimal strongly connected component of $\Gamma(\mathcal{A})$ the proof considers $P = Q \setminus D$. It shows that this set has the proper size (at least $|Q| - k$) and by contradiction shows that $P$ is unreachable.

The combination of Theorem 1.2 and Theorem 1.3 allows to state:


$\Gamma_1(\mathcal{E}_5)$

$\Gamma_2(\mathcal{E}_5)$

$\Gamma_3(\mathcal{E}_5)$

Figure 1.5: The construction of graph $\Gamma(\mathcal{E}_5)$

**Theorem 1.4.** *An automaton $\mathcal{A}$ is completely reachable if and only if the graph $\Gamma(\mathcal{A})$ is strongly connected.*

This theorem gives a theoretical condition when considering completely reachable automata. But Theorems 1.2 and 1.3 give more information and open more considerations than their combination. The first question is if their strict converses are true. For Theorem 1.2 this means to know if having a completely reachable automaton such that any subset of states can be reached with words of defect at most $k$ implies that the graph $\Gamma_k$ is strongly connected. We know from the automaton $\mathcal{P}_4$ given in [21] that this is not the case for $k = 1$, but for bigger values the question is open. In the other case, for Theorem 1.3 it is known a series of automata with $n > 2$ states such that there is a not reachable subset of size $n - 1$ and the construction of $\Gamma$ takes $n - 1$ steps.

Bondar and Volkov in [10] show two series of automata $\mathcal{E}_{n,k}$ and $\mathcal{E}'_{n,k}$with $n$ states such that for both the construction of their respective graphs $\Gamma$ takes $k$ steps and with one the process ends in SUCCESS and with the other in

FAILURE. For both series the alphabet grows with $n$ and $k$, it is still open if the same thing can be done with a series of automata with fixed alphabet size.

## 1.3 Construction of Rystsov graphs

Thanks to Ferens and Szykuła's results in [17] it is possible to decide whether a given automaton is completely reachable in polynomial time. Their algorithm does not make use of Rystsov graphs. Nevertheless it is natural to ask about the practicality of the construction of these graphs. Not only for theoretical purposes, but also because these graphs can provide useful information about the length of the words reaching the corresponding subsets. For instance, the proof of Theorem 1.2 make use of the set of words that enforce the edges present in Rystsov graphs. Also, in Chapter 3 we show how knowing a subset of enforcing words help us to bound the length of the reaching words in a specific case.

From the rest of this section we consider an arbitrary but fixed automaton $\mathcal{A} = \langle Q, \Sigma \rangle$. Recall that $W_k(\mathcal{A})$, for $k \geq 1$, is the set of all the words of defect $k$ of $\mathcal{A}$. In principle the construction of $\Gamma_k(\mathcal{A})$ requires to consider all words in $W_k(\mathcal{A})$. This presents a challenge, since for $k \geq 1$ there are [2]

$$\left\{ {n \atop n-k} \right\} (n-k)!$$

possible transformations of defect $k$ from a set of $n > 1$ elements to itself. This implies that to consider the whole set $W_k(\mathcal{A})$ could have an exponential complexity. Even the seemingly simpler problem of constructing $\Gamma_1(\mathcal{A})$ by this way is, at first sight, impractical.

As previously mentioned, Gonze and Jungers in [21] proposed an algorithm such that given an arbitrary automaton $\mathcal{A}$ with $n \geq 1$ states it constructs the graph $\Gamma_1(\mathcal{A})$ in $O(n^2)$ time. The key observation in their work is that for two words $w$ and $v$ of defect 1 where $\operatorname{excl}(w) = \operatorname{excl}(v)$ and $\operatorname{dupl}(w) = \operatorname{dupl}(v)$, if $u$ is a word such that $wu$ has defect 1, then $vu$ also will have defect 1 and moreover $\operatorname{excl}(wu) = \operatorname{excl}(vu)$ and $\operatorname{dupl}(wu) = \operatorname{dupl}(vu)$. This allows to make a Breadth First Search in the set of the words of defect 1, knowing that once no new words are added then there will be no new edges.

In this section we extend their idea with bigger defects.

[2] Where $\left\{ {n \atop i} \right\}$ is the Stirling number of second kind and it represents the number of ways to partition a set of $n$ elements in $i$ non-empty sets.

The key is, instead of working with the words, we consider the excluded and duplicate sets that this words have. For $k \geq 1$ we define the following set of pairs of subsets:

$$XD_k(\mathcal{A}) := \{(X, D) \mid X = \operatorname{excl}(w) \text{ and } D = \operatorname{dupl}(w) \text{ for some } w \in W_k(\mathcal{A})\}.$$

For the case of $k = 1$, for each edge in the graph $\Gamma_1(\mathcal{A})$ there is a unique pair in $XD_1(\mathcal{A})$. In the case of $k > 1$ it is easy to calculate the edges in $\Gamma_k(\mathcal{A})$ from the pairs in $XD_k(\mathcal{A})$. A pair $(X, D)$ defines an edge in $\Gamma_k(\mathcal{A})$ if there are $C, D \in Q_{k-1}$ such that $X \subseteq \operatorname{leaf}(C)$ and $D \cap \operatorname{leaf}(D) \neq \emptyset$. We shift from working with words, or transformations, to working with pairs of subsets. Let us show a bound to the amount of pairs in $XD_k(\mathcal{A})$ given a fixed $k$ to see this is a significant change.

There are $\binom{n}{k}$ possible sets of size $k$. This accounts for all the possible subsets $X$. The subsets $D$ must be contained in $Q \setminus X$, the latter of size $(n - k)$; besides that, its size ranges from 1 to $\min\{k, n - k\}$. With this considerations at hand we can deduce that

$$|XD_k(\mathcal{A})| \leq \binom{n}{k} \cdot \sum_{d=1}^{\min\{k,n-k\}} \binom{n-k}{d}.$$

In order to give a more precise bound first consider the case when $k \geq n-k$. The second factor of the product is equals to $2^{n-k} - 1 < 2^k$. Thus the bound is $< \binom{n}{k} \cdot 2^k$, which is a polynomial in $n$ with exponent $k$. The case when $k \leq n - k$ is more complicated. We will show by induction on $k$ that the second factor is less than $\binom{n}{k}$. In the case of $k = 1$:

$$\binom{n-1}{1} = n - 1 < n = \binom{n}{1}.$$

Now suppose that $1 < k < n - k$, then

$$\sum_{d=1}^{k} \binom{n-k}{d} = \sum_{d=1}^{k-1} \binom{(n-1)-(k-1)}{d} + \binom{n-k}{k}.$$

The addends of the first sum in the right member of the equality are obtained by adding and subtracting 1 from the top argument $n - k$. Thus, applying the induction hypothesis with $n - 1$ in the role of $n$ we have that

$$\sum_{d=1}^{k} \binom{n-k}{d} < \binom{n-1}{k-1} + \binom{n-k}{k} < \binom{n-1}{k-1} + \binom{n-1}{k} = \binom{n}{k}.$$

We use the fact that binomial coefficient increments when so does its top argument and Pascal's rule. Hence we obtain that in this second case

$$|XD_k(\mathcal{A})| \leq \binom{n}{k}^2.$$

From the previous considerations we can state the following.

**Lemma 1.2.** *For any automata $\mathcal{A}$ and any fixed $k < n$ the cardinality of the set $XD_k(\mathcal{A})$ is upper bounded by a polynomial in $n$ of degree $2k$.*

Although we have successfully diminished the amount of objects to consider, they are still linked to words of the automaton. We need an effective form to calculate the words that produces the aforementioned pairs.

Recall that the *prefix* of a word $w$ is any word $v$ such that $w = vu$. This is, a prefix is an initial sequence of letters. A subset of words $L$ is *prefix-closed* if for each $w \in L$, every prefix of $w$ is also in $L$. For reasons that will be apparent shortly ahead in this section our next step is to prove the existence, for each $k \geq 1$, of a prefix-closed set of words that generates all the pairs in $XD_k(\mathcal{A})$.

For this it is necessary to ensure that for a fixed $v$ the pair $(\mathrm{excl}(uv), \mathrm{dupl}(uv))$ does not depend on the particular prefix $u$. One can say this more explicitly with the next lemma.

**Lemma 1.3.** *For all words $u, u', v \in \Sigma^*$ if*

$$(\mathrm{excl}(u), \mathrm{dupl}(u)) = (\mathrm{excl}(u'), \mathrm{dupl}(u')),$$

*then*

$$(\mathrm{excl}(uv), \mathrm{dupl}(uv)) = (\mathrm{excl}(u'v), \mathrm{dupl}(u'v)).$$

The previous lemma is a direct consequence of equalities (1.1) and (1.2) in Section 1.1.

Let us fix a linear order $\prec$ on the alphabet $\Sigma$, this order can be extended to the *shortlex* order in $\Sigma^*$. For $w, v \in \Sigma^*$ it happens that $w \prec v$ if $|w| < |v|$ or in the case $|w| = |v|$ then $w = u\,a\,w'$, $v = u\,b\,v'$ for some $a, b \in \Sigma$ such that $a \prec b$. For us it is interesting that $(\Sigma^*, \prec)$ is a well-ordered set, every pair of words can be compared, and this order respects concatenation to left and right, if $w \prec w'$ then $u\,w\,v \prec u\,w'\,v$ for any $u, v \in \Sigma^*$. This order eases the definition of our looked for prefix-closed set. For every pair $(X, D) \in XD_k(\mathcal{A})$ let $w_{X,D}$ be the least shortlex word such that $(X, D) = (\mathrm{excl}(w_{X,D}), \mathrm{dupl}(w_{X,D}))$. Moreover, for each $k \geq 1$ let $\overline{W}_k$ be the set of all $w_{X,D}$ words of corresponding defect; for convenience $\overline{W}_0$ contains only the empty word. Additionally we define the subset

$$\overline{W}_{\leq k} := \bigcup_{\ell=0}^{k} \overline{W}_{\ell}.$$

Let us show this last set is prefix-closed.

**Lemma 1.4.** *If $u$ is a prefix of a word in $\overline{W}_k$ and has defect $\ell \leq k$ then $u \in \overline{W}_\ell$.*

*Proof.* Suppose $u$ is a prefix of a word $w \in \overline{W}_k$, the defect of $u$ is $\ell \leq k$ but $u \notin \overline{W}_\ell$. There is a word $u' \in \overline{W}_\ell$ such that

$$(\mathrm{excl}(u), \mathrm{dupl}(u)) = (\mathrm{excl}(u'), \mathrm{dupl}(u'))$$

and, by definition of the set $\overline{W}_\ell$, this means $u' \prec u$. The fact that $u$ is a prefix of $w$ of lesser defect implies there is a non-empty word $v$ such that $w = uv$. Thus, by Lemma 1.3

$$(\mathrm{excl}(w), \mathrm{dupl}(w)) = (\mathrm{excl}(u'v), \mathrm{dupl}(u'v)).$$

Recall that $\prec$ respects concatenation, in our case $u'v \prec uv = w$. What contradicts the definition of $w$ being the least shortlex word who defines the pair $(\mathrm{excl}(w), \mathrm{dupl}(w))$. □

Note that the map $(X, D) \to w_{X,D}$ is a bijection from $XD_k(\mathcal{A})$ to $\overline{W}_k$, hence the cardinals of these sets are equal. This makes $|\overline{W}_k|$ bounded by a polynomial in $n$. We can conclude the same of $|\overline{W}_{\leq k}|$ since is the disjoint finite union of sets with polynomially bounded cardinals. What we need now is an efficient way to obtain $\overline{W}_{\leq k}$.

### A convenient scheme

Now it is time to describe a scheme to generate a prefix-closed set of words with a certain property using Breadth-First Search (BFS). This scheme is fairly known but we have not found a concrete reference to it, thus for convenience we will describe it with some detail. We will work with *lists* of words. These are finite collections whose elements are in a linear order or in some way indexed; we use square brackets [ ] to distinguish them from sets. Besides this distinction we make use of the two following operations. Given two lists $L = [\alpha, \beta, \ldots]$ and $L' = [\alpha', \beta', \ldots]$ we denote the concatenation of them by

$$L \sqcup L' = [\alpha, \beta, \ldots, \alpha', \beta', \ldots].$$

And for any list $L$ and a word $w$, we *append* $w$ to $L$ by making $w$ the last element of $L$.

Given an linearly ordered alphabet $\Sigma$ and a property $\mathfrak{R}$ of the words in $\Sigma^*$, let $W_{\mathfrak{R}}$ be the set of words that satisfy $\mathfrak{R}$. In the case $W_{\mathfrak{R}}$ is finite and prefix-closed we can use the procedure described in Algorithm 1 to compute $W_{\mathfrak{R}}$.

**Algorithm 1** Process to compute a list of words that satisfy a property $\mathfrak{R}$.

```
Input: An ordered alphabet $\Sigma$.
Output: The subset $W_{\mathfrak{R}} \subseteq \Sigma^*$ of words that satisfy $\mathfrak{R}$.
if $\varepsilon$ does not satisfy $\mathfrak{R}$ then
    $W \leftarrow [\ ]$
else
    $W \leftarrow [\varepsilon]$
$I \leftarrow [\ ]$
for $a \in \Sigma$ do
    if $a$ satisfies $\mathfrak{R}$ then
        append $a$ to $W$
        append $a$ to $I$
while $I \neq [\ ]$ do
    $S \leftarrow [\ ]$
    for $w \in I$ do
        for $a \in \Sigma$ do
            if $wa$ satisfies $\mathfrak{R}$ then
                append $wa$ to $S$
    $W \leftarrow W \sqcup S$
    $I \leftarrow S$
return $W$
```

**Proposition 1.2.** *Let $\Sigma$ be a linearly ordered alphabet with $m$ letters. Let $\mathfrak{R}$ be a property of the words in $\Sigma$ such that:*

1. *to check whether an arbitrary word $w \in \Sigma^*$ satisfies $\mathfrak{R}$ takes time $\leq T$;*
2. *the subset $W_{\mathfrak{R}} \subseteq \Sigma^*$ of words satisfying $\mathfrak{R}$ is prefix-closed and*
3. *$|W_{\mathfrak{R}}| \leq N$, for a positive integer $N$.*

*The procedure described in Algorithm 1 returns the list $W_{\mathfrak{R}}$ in ascending shortlex order in time $\leq mNT$.*

*Proof.* First we prove that the set of words returned by Algorithm 1 is shortlex ordered. The words produced on the $\ell$ round of the **while** are of the form $wa$ for $w \in I$ and $a \in \Sigma$. The word $w$ is produced in the round $\ell - 1$. An argument by induction on $\ell$ allows us to conclude that all the words appended to $S$ have the same length, this is $\ell$. Now, suppose that in a certain round the words $w, w' \in I$ are processed, moreover suppose that $w \prec w'$. For any $a \in \Sigma$ the word $wa$ is processed before $w'a$, even more if $a, b \in \Sigma$ and $a \prec b$, the word $wa$ is processed before than $wb$. Altogether, this allows us to conclude that the list $W$ is constructed in shortlex order.

Second, we prove Algorithm 1 returns exactly $W_{\mathfrak{R}}$. The fact that the returned set $W$ is a subset of $W_{\mathfrak{R}}$ comes from the definition of the algorithm. We need to prove that if $w \in W_{\mathfrak{R}}$ then is added eventually to $W$. We do it by induction on length $|w|$. If $\varepsilon$ satisfies $\mathfrak{R}$ we know it is included in $W$. If not, at least one letter satisfies the property thus the words of length 1 are included. In the extreme case $W_{\mathfrak{R}} = [\ ]$ the algorithm still returns the correct set. Recall the condition of being prefix-closed, thus every $w \in W_{\mathfrak{R}}$ with $|w| > 1$ has the form $w'a$ with $a \in \Sigma$ and $w' \in W_{\mathfrak{R}}$. By the induction hypothesis $w' \in W$, hence $w$ is appended to $W$ in the round $|w|$.

Finally, the procedure verifies whether or not $wa$ satisfies $\mathfrak{R}$ for every pair $(w, a) \in W_{\mathfrak{R}} \times \Sigma$. There are $\leq Nm$ of these pairs. Additionally the procedure takes $\leq T$ time to check each pair. Hence the time is $\leq mNT$. □

## The computation of the graph

Once described the general scheme to compute a set of words we are set to finally compute the Rystsov graph as it was our initial intention. First order the alphabet by defect of the letters, with an arbitrary order among letters with the same defect. In our case the property to use in Algorithm 1 is "$w \in \overline{W}_{\leq k}$" or, in other words "*$w$ is the least shortlex word such that* $(\mathrm{excl}(w), \mathrm{dupl}(w)) \in XD_\ell(\mathcal{A})$ *for* $\ell \leq k$." Lemma 1.4 tells us $\overline{W}_{\leq k}$ is prefix closed and we already know its cardinal is $O(n^{2k})$ bounded.

What is left is to bound the time $T$ needed check our property. In order to make this we need to make some minor modifications and considerations to the scheme. In our case the lists involved will not contain just mere words but triplets of the form $(w, \mathrm{excl}(w), \mathrm{dupl}(w))$. An appropriate data structure for storing the lists will helps us to reduce the time of look-up and insertion tasks; in this case self-balancing binary search trees allow these operations to be made in a logarithmic time with respect to the size of the lists. Thus, the checking of the property will be made in two phases for each $(w, \mathrm{excl}(w), \mathrm{dupl}(w)) \in I$ and $a \in \Sigma$:

(A) Compute $X := \mathrm{excl}(wa)$ and $D := \mathrm{dupl}(wa)$ to form the triplet $(wa, \mathrm{excl}(wa), \mathrm{dupl}(wa))$.

(B) Verify if the part $(X, D)$ is not already present in a triplet in $W$ or $S$.

For (A) we make use of equalities (1.1) and (1.2) in Subsection 1.1. They just require to know $\mathrm{excl}(w)$, $\mathrm{dupl}(w)$ and $q \cdot a^{-1}$ for $q \in Q$; the first two already given and the latter can be computed beforehand once, for every $a \in \Sigma$, and adequately stored to be used when it is necessary. Recall that the sizes of the excluded and duplicate sets are at most $k \leq n$, then this operation can be done in $O(\log(n))$ time.

The execution of (B) requires a look-up in lists of size at most $O(n^{2k})$, and as we already established this can be done in logarithmic time of this size, thus we still require $O(\log(n))$. To verify (B) is enough to determine if $wa$ satisfies our looked for property. Proposition 1.2 and its proof tell us the lists are created in ascending shortlex order. Thus, if the sets $(X, D)$ are not part of any triplet in $W$, then $wa$ is the shortest word such that $(\mathrm{excl}(wa), \mathrm{dupl}(wa)) = (X, D)$. Furthermore, since the concatenation of the letters is done in the linear order, if for two letters $a, b \in \Sigma$ with $a \prec b$ it happens that $\mathrm{excl}(wa) = \mathrm{excl}(wb)$ and $\mathrm{dupl}(wa) = \mathrm{dupl}(wb)$, the word $wa$ will be added to $S$ before and $wa \prec wb$ as we wanted.

Now we have all the components to assemble the Rystsov graph $\Gamma_k(\mathcal{A})$; assuming we have the vertex set $Q_{k-1}$. Once obtained the list $W$ from Algorithm 1, we extract the triplets $(w, X, D)$ such that $|X| = k$. As we explained before we can construct the edges of $\Gamma_k(\mathcal{A})$ from these pairs. The process of verifying that $X \subset \mathrm{leaf}(C)$ and $D \cap \mathrm{leaf}(C') \neq \emptyset$ for a pair of vertex $C, C' \in Q_{k-1}$ can be done in time $O(\log(n))$ since $k$ is fixed. And we need to check this at most $|XD_k(\mathcal{A})| = O(n^{2k})$ times. Hence the computation of the edges is done in time $O(n^{2k}\log(n))$. In conclusion we can summarize this as follows.

**Theorem 1.5.** *Let $\mathcal{A}$ be an automaton with $n$ states and $m$ input letters. For each $k < n$, there exists an algorithm that builds the graph $\Gamma_k(\mathcal{A})$ in time $O(mn^{2k}\log(n))$.*

# Chapter 2

# Binary completely reachable automata

Here we study completely reachable automata with two input letters; for brevity, we call these kind of automata *binary.* Our main results provide a new characterization of binary completely reachable automata, and the characterization leads to a quasilinear time algorithm for recognizing complete reachability for binary automata. This chapter was based on [11].

The main question of this chapter is: *under which conditions is a binary automaton completely reachable?* The rest of the section presents a series of reductions showing that to answer this question, it suffices to analyse automata of a specific form.

Let $\mathcal{A} = \langle Q, \{a, b\}\rangle$ be a binary automaton with $n > 1$ states. If neither $a$ nor $b$ has defect 1, no subset of size $n-1$ is reachable in $\mathcal{A}$. Therefore, when looking for binary completely reachable automata, we must focus on automata possessing a letter of defect 1. From now on we will denote the necessary letter of defect 1 as $a$.

The image of every non-empty word over $\{a, b\}$ is contained in either $Q \cdot a$ or $Q \cdot b$. If the defect of $b$ is greater than or equal to 1, then at most two subsets of size $n-1$ are reachable (namely, $Q \cdot a$ and $Q \cdot b$), whence $\mathcal{A}$ can only be completely reachable provided that $n = 2$. The automaton $\mathcal{A}$ is then nothing but the classical flip-flop, see Figure 2.1.

Having isolated this exception, we assume from now on that $n \geq 2$ and the letter $b$ has defect 0, which means that $b$ acts as a permutation of $Q$. The following fact was first stated in [9] without proof.

**Lemma 2.1.** *If $\mathcal{A} = \langle Q, \{a, b\}\rangle$ is a completely reachable automaton in which the letter $b$ acts as a permutation of $Q$, then $b$ acts as a cyclic permutation.*

*Proof.* The argument will be by contradiction. Suppose that the cyclic de-

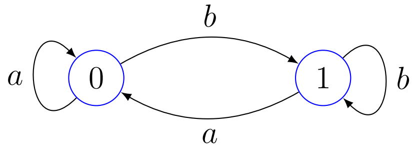


Figure 2.1: The flip-flop automaton.

composition of the permutation $b$ has $k \geq 2$ cycles $\pi_1, \ldots, \pi_k$. For each $1 \leq i \leq k$ denote by $Q_i$ the set of states moved by $\pi_i$. Note that this collection of subsets is a partition of $Q$, because they are disjoint from each other and $Q = \bigcup_{i=1}^{k} Q_i$. Since $a$ is of defect 1, there is exactly one $Q_i$ such that $Q_i \not\subset Q.a$. The automaton $\mathcal{A}$ is completely reachable, therefore there is a word $w \in \{a, b\}^*$ such that $Q \cdot w = Q_i$. The subset $Q_i \neq Q$, thus $|w| > 0$. Additionally, we will choose the word $w$ to be the shortest that reaches $Q_i$. Now consider the rightmost letter of $w$. First suppose that $w = w'b$ for some $w' \in \{a, b\}^*$. If $m \geq 1$ is the least common multiple of the lengths of the cycles $\pi_1, \ldots, \pi_k$, then $b^m$ is the identity transformation. We use this fact when applying $b^{m-1}$ to the equality

$$Q \cdot w = Q \cdot w'b = Q_i,$$

obtaining

$$Q \cdot wb^{m-1} = Q \cdot w' = Q_i b^{m-1}.$$

Applying $b^{m-1}$ to $Q_i$ lets the subset unchanged, thus we get $Q \cdot w' = Q_i$. The last equality contradicts our choice of $w$ being the shortest to reach $Q_i$, therefore $b$ can not be the last letter of $w$. Then, the last letter of $w$ is $a$. Note that $Q \cdot w \subseteq Q \cdot a$, what contradicts the fact that $Q_i \not\subseteq Q \cdot a$. This contradiction came from assuming the cyclic decomposition of $b$ has more than 1 cycle. □

Taking Lemma 2.1 into account, we restrict our further considerations to automata with $n > 2$ states and two input letters $a$ and $b$ such that $a$ has defect 1 and $b$ acts as a cyclic permutation of $Q$. Without any loss, we will additionally assume that these automata have the set $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$ of all residues modulo $n$ as their state set and the action of $b$ at any state merely adds 1 modulo $n$. Let us also agree that whenever we deal with elements of $\mathbb{Z}_n$ the signs $\oplus$ and $\ominus$ mean addition and subtraction modulo $n$ and the signs $+$ and $-$ the usual addition and subtraction.

Further, we will assume that $0 = \text{excl}(a)$ as it does not matter from which origin the cyclic count of the states start.

Since $b$ is a permutation, for each $k \in \mathbb{Z}_n$, the transformations $q \mapsto q \cdot b^k a$ and $q \mapsto q \cdot b$ generate the same submonoid in the monoid of all transformations of $\mathbb{Z}_n$ as do the transformations $q \mapsto q \cdot a$ and $q \mapsto q \cdot b$. This means that if one treats the word $b^k a$ as a new letter $a_k$, say, one gets the automaton $\mathcal{A}_k = \langle \mathbb{Z}_n, \{a_k, b\} \rangle$ that is syntactically equivalent to $\mathcal{A}$. Therefore, $\mathcal{A}$ is completely reachable if and only if so is $\mathcal{A}_k$ for some (and hence for all) $k$. Hence we may choose $k$ as we wish and study the automaton $\mathcal{A}_k$ for the specified value of $k$ instead of $\mathcal{A}$.

What can we achieve using this? From (1.1) we have $\text{excl}(b^k a) = \text{excl}(a) = 0$. Further, let $q_1 \neq q_2$ be such that $q_1 \cdot a = q_2 \cdot a = \text{dupl}(a)$. Choosing $k = q_1$ (or $k = q_2$), we get $0 \cdot b^k a = \text{dupl}(a)$. Thus, we will assume that $0 \cdot a = \text{dupl}(a)$.

Summarizing, we will consider automata $\langle \mathbb{Z}_n, \{a, b\} \rangle$ such that:

- the letter $a$ has defect 1, $\text{excl}(a) = 0$, and $0 \cdot a = \text{dupl}(a)$;
- $q \cdot b = q \oplus 1$ for each $q \in \mathbb{Z}_n$.

We call such automata *standardized.* For the purpose of complexity considerations at the end of Sect. 2.3, observe that given a binary automaton $\mathcal{A}$ in which one letter acts as a cyclic permutation while the other has defect 1, one can 'standardize' the automaton, that is, construct a standardized automaton syntactically equivalent to $\mathcal{A}$, in linear time with respect to the size of $\mathcal{A}$. However this change is not innocuous. In Chapter 3 we see how in order to bound the length of the reaching words of proper subsets there is a difference whether the automaton is standardized or not.

## 2.1 A necessary condition

Let $\langle \mathbb{Z}_n, \{a, b\} \rangle$ be a standardized automaton and $w \in \{a, b\}^*$. A subset $S \subseteq \mathbb{Z}_n$ is said to be *w-invariant* if $S \cdot w \subseteq S$.

**Proposition 2.1.** *If $\langle \mathbb{Z}_n, \{a, b\} \rangle$ is a completely reachable standardized automaton, then no proper subgroup of $(\mathbb{Z}_n, \oplus)$ is a-invariant.*

*Proof.* Arguing by contradiction, assume that $H \subsetneqq \mathbb{Z}_n$ is a subgroup such that $H \cdot a \subseteq H$. Let $d$ stand for the index of the subgroup $H$ in the group $(\mathbb{Z}_n, \oplus)$. The set $\mathbb{Z}_n$ is then partitioned into the $d$ cosets

$$H_0 = H,\ H_1 = H \cdot b = H \oplus 1,\ \ldots,\ H_{d-1} = H \cdot b^{d-1} = H \oplus d - 1.$$

For $i = 0, 1, \ldots, d-1$, let $T_i$ be the complement of the coset $H_i$ in $\mathbb{Z}_n$. Then we have $T_i = \cup_{j \neq i} H_j$ and $T_i \cdot b = T_{i \oplus 1 \,(\text{mod } d)}$ for each $i = 0, 1, \ldots, d-1$.

Since $\mathcal{A}$ is completely reachable, each subset $T_i$ is reachable. Take a word $w$ of minimum length among words with the image equal to one of the subsets $T_0, T_1, \dots, T_{d-1}$. Write $w$ as $w = w'c$ for some letter $c \in \{a, b\}$.

If $c = b$, then for some $i \in \{0, 1, \dots, d-1\}$, we have

$$\mathbb{Z}_n \cdot w'b = T_i = T_{i-1 \pmod d} \cdot b.$$

Since $b^n$ acts as the identity mapping, applying the word $b^{n-1}$ to this equality yields $\mathbb{Z}_n \cdot w' = T_{i-1 (\text{mod } d)}$ whence the image of $w'$ is also equal to one of the subsets $T_0, T_1, \dots, T_{d-1}$. This contradicts the choice of $w$.

Thus, $c = a$, whence the set $\mathbb{Z}_n \cdot w$ is contained in $\mathbb{Z}_n \cdot a$. The only $T_i$ that is contained in $\mathbb{Z}_n \cdot a$ is $T_0$ because each $T_i$ with $i \neq 0$ contains $H_0$, and $H_0 = H$ contains 0, the excluded state of $a$. Hence, $\mathbb{Z}_n \cdot w = T_0$, that is, $\mathbb{Z}_n \cdot w'a = T_0$. For each state $q \in \mathbb{Z}_n \cdot w'$, we have $q \cdot a \in T_0$, and this implies $q \in T_0$ since $H_0$, the complement of $T_0$, is $a$-invariant. We see that $\mathbb{Z}_n \cdot w' \subseteq T_0$ and the inclusion cannot be strict because $T_0$ cannot be the image of its proper subset. However, the equality $\mathbb{Z}_n \cdot w' = T_0$ again contradicts the choice of $w$. □

We will show that the condition of Proposition 2.1 is not only necessary but also sufficient for complete reachability of a standardized automaton. The proof of sufficiency will make use of Rystsov graphs. We describe the structure of these graphs in the following sections.

## 2.2 Rystsov graph of a binary automaton

Recall that given a (not necessarily binary) automaton $\mathcal{A} = \langle Q, \Sigma \rangle$, $W_1(\mathcal{A})$ stand for the set of all words in $\Sigma^*$ that have defect 1 in $\mathcal{A}$. Also recall the digraph $\Gamma_1(\mathcal{A})$ with the vertex set $Q$ and the edge set

$$E := \{(\text{excl}(w), \text{dupl}(w)) \mid w \in W_1(\mathcal{A})\}.$$

The sufficient condition for complete reachability from [9] stated in Theorem 1.1 tell us that $\mathcal{A}$ is completely reachable if $\Gamma_1(\mathcal{A})$. In the same article was shown that the condition of Theorem 1.1 is not necessary for complete reachability, but it was conjectured that the condition might characterize binary completely reachable automata. However, this conjecture has been refuted in [8, Example 2] by exhibiting a binary completely reachable automaton with 12 states whose Rystsov graph is not strongly connected. Here we include a similar example which we will use to illustrate some of our results.

Consider the standardized DFA $\mathcal{E}'_{12} = \langle \mathbb{Z}_{12}, \{a, b\} \rangle$ where the action of the letter $a$ is specified as follows:

| $q$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $q \cdot a$ | 10 | 1 | 2 | 8 | 4 | 5 | 10 | 9 | 3 | 7 | 6 | 11 |

(The automaton $\mathcal{E}'_{12}$ only slightly differs from the automaton $\mathcal{E}_{12}$ used in [8, Example 2], hence the notation.) The automaton $\mathcal{E}'_{12}$ is shown in Figure 2.2, in which we have replaced edges that should have been labeled $a$ and $b$ with solid and, resp., dashed edges.

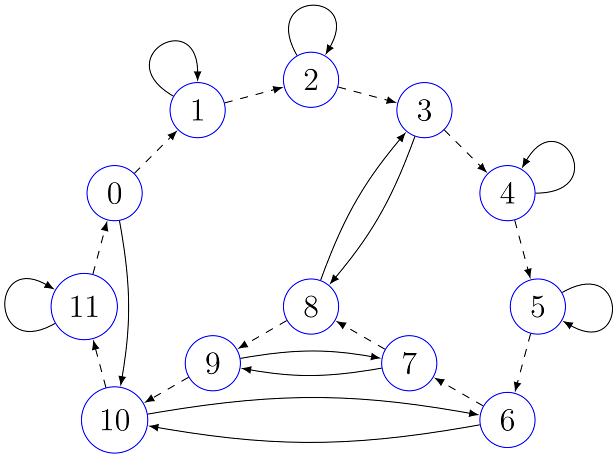


Figure 2.2: The automaton $\mathcal{E}'_{12}$; solid and dashed edges show the action of $a$ and, resp., $b$

We postpone the description of the digraph $\Gamma_1(\mathcal{E}'_{12})$ and the proof that the automaton $\mathcal{E}'_{12}$ is completely reachable until we develop suitable tools that make the description and the proof easy.

We start with a characterization of Rystsov graphs of standardized automata. Let $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\} \rangle$ be such a automaton. It readily follows from (1.1) and (1.2) that $\operatorname{excl}(w) \cdot b = \operatorname{excl}(wb)$ and $\operatorname{dupl}(w) \cdot b = \operatorname{dupl}(wb)$ for every word $w \in W_1(\mathcal{A})$. Therefore, the edge set $E$ of the digraph $\Gamma_1(\mathcal{A})$ is closed under the *translation* $(q, p) \mapsto (q \cdot b, p \cdot b) = (q \oplus 1, p \oplus 1)$. As a consequence, for any edge $(q, p) \in E$ and any $k \in \mathbb{Z}_n$, the pair $(q \oplus k, p \oplus k)$ also constitutes an edge in $E$.

Denote by $D_1(\mathcal{A})$ the set of ends of edges of $\Gamma_1(\mathcal{A})$ that start at 0, that is,

$$D_1(\mathcal{A}) := \{p \in \mathbb{Z}_n \mid 0 \to p \in E\}.$$

We call $D_1(\mathcal{A})$ the *difference set* of $\mathcal{A}$. Our first observation shows how to recover all edges of $\Gamma_1(\mathcal{A})$, knowing $D_1(\mathcal{A})$.

**Lemma 2.2.** *Let $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\}\rangle$ be a standardized automaton. A pair $(q, p) \in \mathbb{Z}_n \times \mathbb{Z}_n$ forms an edge in the digraph $\Gamma_1(\mathcal{A})$ if and only if $p \ominus q \in D_1(\mathcal{A})$.*

*Proof.* If $p \ominus q \in D_1(\mathcal{A})$, the pair $(0, p \ominus q)$ is an edge in $E$, and therefore, so is the pair $(0 \oplus q, (p \ominus q) \oplus q) = (q, p)$. Conversely, if $(q, p)$ is an edge in $E$, then so is $(q \oplus (n \ominus q), p \oplus (n \ominus q)) = (0, p \ominus q)$, whence $p \ominus q \in D_1(\mathcal{A})$. □

By Lemma 2.2, the presence or absence of an edge in $\Gamma_1(\mathcal{A})$ depends only on the difference modulo $n$ of two vertex numbers. This means that $\Gamma_1(\mathcal{A})$ is a *circulant* digraph, that is, the Cayley digraph of the cyclic group $(\mathbb{Z}_n, \oplus)$ with respect to some subset of $\mathbb{Z}_n$. Recall the notion of *Cayley digraph* presented in Definition 1.2. Let $H_1(\mathcal{A})$ stand for the subgroup of the group $(\mathbb{Z}_n, \oplus)$ generated by the difference set $D_1(\mathcal{A})$. Specializing Lemma 1.1 to this case, we get the following description for Rystsov graphs of standardized automata. Even more we get a sure way to determine beforehand if we will get a strongly connected graph $\Gamma_1(\mathcal{A})$.

**Proposition 2.2.** *Let $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\}\rangle$ be a standardized automaton. The digraph $\Gamma_1(\mathcal{A})$ is isomorphic to the Cayley digraph $\mathrm{Cay}(\mathbb{Z}_n, D_1(\mathcal{A}))$. The strongly connected components of $\Gamma_1(\mathcal{A})$ have the cosets of the subgroup $H_1(\mathcal{A})$ as their vertex sets, and each strongly connected component is isomorphic to the Cayley digraph $\mathrm{Cay}(H_1(\mathcal{A}), D_1(\mathcal{A}))$. In particular, the digraph $\Gamma_1(\mathcal{A})$ is strongly connected if and only if the set $D_1(\mathcal{A})$ generates $(\mathbb{Z}_n, \oplus)$ or, equivalently, if and only if the greatest common divisor of $D_1(\mathcal{A})$ is coprime to $n$.*

Proposition 2.2 shows that structure of the Rystsov graph of a standardized automaton $\mathcal{A}$ crucially depends on its difference set $D_1(\mathcal{A})$. The definition of the edge set of $\Gamma_1(\mathcal{A})$ describes $D_1(\mathcal{A})$ as the set of duplicate states for all words $w$ of defect 1 whose excluded state is 0, that is,

$$D_1(\mathcal{A}) = \{\mathrm{dupl}(w) \mid \mathrm{excl}(w) = 0\}.$$

Thus, understanding of difference sets amounts to a classification of transformations caused by words of defect 1. It is such a classification that is behind the handy description of difference sets stated in Proposition 2.3. But before this, we need the following lemma. It allows us find the exact transformations that preserve the image of the letter $a$.

**Lemma 2.3.** *Let $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\}\rangle$ be a standardized automaton. Denote by $N$ the image of the letter $a$, i.e., $N := \mathbb{Z}_n \backslash \{0\}$. Thus both the transformations $b^r a$ and $a$ act as permutations on the set $N$.*

*Proof.* If $q \cdot a = p$ for some $q \in \mathbb{Z}_n$ and $p \in N$, then, clearly, $(q \ominus r) \cdot b^r a = p$. Hence the only state in $N$ that has a preimage of size 2 under the actions of both $a$ and $b^r a$ is

$$\mathrm{dupl}(a) = \begin{cases} 0 \cdot a = r \cdot a, \\ (n \ominus r) \cdot b^r a = 0 \cdot b^r a, \end{cases}$$

and in both cases 0 belongs to the preimage. Thus, the preimage of every $p \in N$ under both $a$ and $b^r a$ contains a unique state in $N$, which means that both $a$ and $b^r a$ act on the set $N$ as permutations. □

**Proposition 2.3.** *Let $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\} \rangle$ be a standardized automaton. Let $r \neq 0$ be such that $r \cdot a = \mathrm{dupl}(a)$. Then*

$$D_1(\mathcal{A}) = \{\mathrm{dupl}(a) \cdot v \mid v \in \{a, b^r a\}^*\}. \tag{2.1}$$

*Proof.* Denote by $N$ the image of the letter $a$. Lemma 2.3 states that every word $v \in \{a, b^r a\}^*$ acts as a permutation on $N$. Then the word $av$ has defect 1 and $\mathrm{excl}(av) = 0$. Applying the equality (1.2) with $a$ in the role of $u$, we derive that $\mathrm{dupl}(av) = \mathrm{dupl}(a) \cdot v$. Thus, denoting the right-hand side of (2.1) by $D$, we see that every state in $D$ is the duplicate state of some word whose only excluded state is 0. This means that $D_1(\mathcal{A}) \supseteq D$.

To verify the converse inclusion, take an arbitrary state $p \in D_1(\mathcal{A})$ and let $w$ be a word of defect 1 such that $\mathrm{excl}(w) = 0$ and $\mathrm{dupl}(w) = p$. Since $\mathrm{excl}(w) = 0$, the word $w$ ends with the letter $a$[1]. We prove that $p$ lies in $D$ by induction on the number of occurrences of $a$ in $w$. If $a$ occurs in $w$ once, then $w = b^k a$ for some $k \in \mathbb{Z}_n$. We have $p = \mathrm{dupl}(w) = \mathrm{dupl}(b^k a) = \mathrm{dupl}(a) \in D$.

If $a$ occurs in $w$ at least twice, write $w = w' b^k a$ where $w'$ ends with $a$. Then the word $w'$ has defect 1 and $\mathrm{excl}(w') = 0$. As $w'$ has fewer occurrences of $a$, the inductive assumption applies and yields $\mathrm{dupl}(w') \in D$. If $\mathrm{dupl}(w') := p'$, we have $p = p' \cdot b^k a$. If we prove that $k \in \{0, r\}$, we are done since the set $D$ is both $a$-invariant and $b^r a$-invariant by its definition. Arguing by contradiction, assume $k \notin \{0, r\}$. Let $\ell = k \cdot a$; then $k$ is the only state in $\ell a^{-1}$. Hence $\ell a^{-1} = \mathrm{excl}(w' b^k)$, and the equality (1.1) (with $u = w' b^k$ and $v = a$) shows that $\ell \in \mathrm{excl}(w' b^k a) = \mathrm{excl}(w)$. Clearly, $\ell \neq 0$ as $\ell$ lies in the image of $a$. Therefore the conclusion $\ell \in \mathrm{excl}(w)$ contradicts the assumption $\mathrm{excl}(w) = 0$. □

For an illustration, we apply (2.1) to compute the difference set for the automaton $\mathcal{E}'_{12}$ shown in Fig. 2.2. In $\mathcal{E}'_{12}$, we have $r = 6$ and $\mathrm{dupl}(a) = 10$.

[1] It could also happen that $w = w' a (b^n)^i$ with $i \geq 1$, $w' a$ of defect 1 and $\mathrm{dupl}(w' a) = p$. But, since $b^n$ acts as the identity in $\mathbb{Z}_n$ it is not worthy to consider this case.

Acting by $a$ and $b^6a$ gives $10 \cdot a = 6$ and $10 \cdot b^6 a = (10 \oplus 6) \cdot a = 4 \cdot a = 4$. Thus, $4, 6 \in D_1(\mathcal{E}'_{12})$. Acting by $a$ or $b^6a$ at 4 and 6 does not produce anything new: $4 \cdot a = 4$ and $4 \cdot b^6 a = (4 \oplus 6) \cdot a = 10 \cdot a = 6$ while $6 \cdot a = 10$ and $6 \cdot b^6 a = (6 \oplus 6) \cdot a = 0 \cdot a = 10$. We conclude that $D_1(\mathcal{E}'_{12}) = \{4, 6, 10\}$. Since 2, the greatest common divisor of $\{4, 6, 10\}$, divides 12, we see that the digraph $\Gamma_1(\mathcal{E}'_{12})$ is not strongly connected. The subgroup $H_1(\mathcal{E}'_{12})$ consists of even residues modulo 12 and has index 2. Hence the digraph $\Gamma_1(\mathcal{E}'_{12})$ has two strongly connected components whose vertex sets are $\{0, 2, 4, 6, 8, 10\}$ and $\{1, 3, 5, 7, 9, 11\}$, and for each $q \in \mathbb{Z}_{12}$, it has the edges $(q, q \oplus 4)$, $(q, q \oplus 6)$, and $(q, q \oplus 10)$.

In fact, formula (2.1) leads to a straightforward algorithm that computes the difference set of any standardized automaton $\mathcal{A}$ in time linear in $n$. This, together with Proposition 2.2, gives an efficient way to compute the Rystsov graph of $\mathcal{A}$.

Let $D_1^0(\mathcal{A}) = D_1(\mathcal{A}) \cup \{0\}$. It turns out that $D_1^0(\mathcal{A})$ is always a union of cosets of a nontrivial subgroup.

**Proposition 2.4.** *Let $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\} \rangle$ be a standardized automaton. Let $r \neq 0$ be such that $r \cdot a = \mathrm{dupl}(a)$. Then the set $D_1^0(\mathcal{A})$ is a union of cosets of the subgroup generated by $r$ in the group $H_1(\mathcal{A})$.*

*Proof.* It is easy to see that the claim is equivalent to the following implication: if $d \in D_1^0(\mathcal{A})$, then $d \oplus r \in D_1^0(\mathcal{A})$. This clearly holds if $d \oplus r = 0$. Thus, assume that $d \in D_1^0(\mathcal{A})$ is such that $d \oplus r \neq 0$. Then $(d \oplus r) \cdot a \in D_1(\mathcal{A})$. Indeed, if $d = 0$, then $(d \oplus r) \cdot a = r \cdot a = \mathrm{dupl}(a) \in D_1(\mathcal{A})$. If $d \neq 0$, then $d \in D_1(\mathcal{A})$, whence $(d \oplus r) \cdot a = d \cdot b^r a \in D_1(\mathcal{A})$ as formula (2.1) ensures that the set $D_1(\mathcal{A})$ is closed under the action of the word $b^r a$.

Thanks to Lemma 2.3 we know that $a$ acts on the set $N = \mathbb{Z}_n \backslash \{0\}$ as a permutation. Hence for some $k$, the word $a^k$ acts on $N$ as the identity map. Then $d \oplus r = (d \oplus r) \cdot a^k = ((d \oplus r) \cdot a) \cdot a^{k-1} \in D_1(\mathcal{A})$ since we have already shown that $(d \oplus r) \cdot a \in D_1(\mathcal{A})$ and formula (2.1) ensures that the set $D_1(\mathcal{A})$ is $a$-invariant. □

In our running example $\mathcal{E}'_{12}$, $r = 6$ and the set $D_1^0(\mathcal{E}'_{12}) = \{0, 4, 6, 10\}$ is the union of the subgroup $\{0, 6\}$ with its coset $\{4, 10\}$ in the group $H_1(\mathcal{E}'_{12})$.

Let $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\} \rangle$ be a standardized automaton. Proposition 2.4 shows that then the set $D_1^0(\mathcal{A})$ is situated between the subgroup $H_1(\mathcal{A})$ and the subgroup $R$ generated by $r \neq 0$ such that $r \cdot a = \mathrm{dupl}(a)$:

$$R \subseteq D_1^0(\mathcal{A}) \subseteq H_1(\mathcal{A}). \tag{2.2}$$

Formula (2.1) implies that the difference set $D_1(\mathcal{A})$ is $a$-invariant, and so is the set $D_1^0(\mathcal{A})$ since $0 \cdot a = \mathrm{dupl}(a) \in D_1(\mathcal{A})$. By Proposition 2.1, if the automaton $\mathcal{A}$ is completely reachable, then either $H_1(\mathcal{A}) = \mathbb{Z}_n$ or $H_1(\mathcal{A})$ is a proper subgroup and both inclusions in (2.2) are strict. Recall that by Proposition 2.2 $H_1(\mathcal{A}) = \mathbb{Z}_n$ if and only if the digraph $\Gamma_1(\mathcal{A})$ is strongly connected. In the other case, $n$ must be a product of at least three (not necessarily distinct) prime numbers. Indeed, the subgroups of $(\mathbb{Z}_n, \oplus)$ ordered by inclusion are in a 1-1 correspondence to the divisors of $n$ ordered by division, and no product of only two primes can have two different proper divisors $d_1$ and $d_2$ such that $d_1$ divides $d_2$. We thus arrive at the following conclusion.

**Corollary 2.1.** *A binary automaton $\mathcal{A}$ with $n$ states where $n$ is a product of two prime numbers is completely reachable if and only if one of its letters acts as a cyclic permutation of the state set, the other letter has defect* 1*, and the digraph $\Gamma_1(\mathcal{A})$ is strongly connected.*

Corollary 2.1 allows one to show that the number of states in a binary completely reachable automata whose Rystsov graph is not strongly connected is at least 12. (Thus, our examples of such automata ($\mathcal{E}_{12}$ from [8, Example 2] and $\mathcal{E}'_{12}$) are of minimum possible size.) Indeed, Corollary 2.1 excludes all sizes less than 12 except 8. If a standardized automaton $\mathcal{A}$ has 8 states and the digraph $\Gamma_1(\mathcal{A})$ is not strongly connected, then the group $H_1(\mathcal{A})$ has size at most 4 and its subgroup $R$ generated by the non-zero state in $\mathrm{dupl}(a)a^{-1}$ has size at least 2. By Proposition 2.4 the set $D_1^0(\mathcal{A})$ is a union of cosets of the subgroup $R$ in the group $H_1(\mathcal{A})$, whence either $D_0(\mathcal{A}) = R$ or $D_0(\mathcal{A}) = H_1(\mathcal{A})$. In either case, we get a proper $a$-invariant subgroup, and Proposition 2.1 implies that the automaton $\mathcal{A}$ is not completely reachable.

## 2.3 Subgroup sequences for standardized automata

As we previously stated, in [10, 8] Theorem 1.1 is generalized in the following way. A sequence of digraphs $\Gamma_1(\mathcal{A})$, $\Gamma_2(\mathcal{A})$, …, $\Gamma_k(\mathcal{A})$, … is assigned to an arbitrary (not necessarily binary) automaton $\mathcal{A}$, where $\Gamma_1(\mathcal{A})$ is the Rystsov graph of $\mathcal{A}$ while the 'higher level' digraphs $\Gamma_2(\mathcal{A})$, …, $\Gamma_k(\mathcal{A})$, … are defined via words that have defect 2, …, $k$, … in $\mathcal{A}$. The length of the sequence is less than the number of states of $\mathcal{A}$, and $\mathcal{A}$ is completely reachable if and only if the final digraph in the sequence is strongly connected.

For the case when $\mathcal{A}$ is a standardized automaton, Proposition 2.2 shows that the Rystsov graph $\Gamma_1(\mathcal{A})$ is completely determined by the difference set

$D_1(\mathcal{A})$ and the subgroup $H_1(\mathcal{A})$ that $D_1(\mathcal{A})$ generates. This suggests that for binary automata, one may substitute the 'higher level' digraphs of [10, 8] by suitably chosen 'higher level' difference sets and their generated subgroups.

Take a standardized automaton $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\}\rangle$ and for each $k > 1$, inductively define the set $D_k(\mathcal{A})$ and the subgroup $H_k(\mathcal{A})$:

$$\begin{aligned} D_k(\mathcal{A}) = \{p \in \mathbb{Z}_n \mid\ & p \in \mathrm{dupl}(w) \text{ for some } w \in \{a, b\}^* \\ & \text{such that } 0 \in \mathrm{excl}(w) \subseteq H_{k-1}(\mathcal{A}),\ |\mathrm{excl}(w)| \le k\}, \\ H_k(\mathcal{A}) \text{ is the}&\text{ subgroup of } (\mathbb{Z}_n, \oplus) \text{ generated by } D_k(\mathcal{A}). \end{aligned} \tag{2.3}$$

Observe that if we let $H_0(\mathcal{A}) = \{0\}$, the definition (2.3) makes sense also for $k = 1$ and leads to exactly the same $D_1(\mathcal{A})$ and $H_1(\mathcal{A})$ as defined in Sect. 2.2.

Using the definition (2.3), it is easy to prove by induction that $D_k(\mathcal{A}) \subseteq D_{k+1}(\mathcal{A})$ and $H_k(\mathcal{A}) \subseteq H_{k+1}(\mathcal{A})$ for all $k$.

**Proposition 2.5.** *If $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\}\rangle$ is a standardized automaton and $H_\ell(\mathcal{A}) = \mathbb{Z}_n$ for some $\ell$, then $\mathcal{A}$ is a completely reachable automaton.*

*Proof.* As $\mathcal{A}$ is fixed, we write $D_k$ and $H_k$ instead of $D_k(\mathcal{A})$ and, resp., $H_k(\mathcal{A})$.

Take any non-empty subset $S \subseteq \mathbb{Z}_n$. We prove that $S$ is reachable in $\mathcal{A}$ by induction on $n - |S|$. If $n - |S| = 0$, there is nothing to prove as $S = \mathbb{Z}_n$ is reachable via the empty word. Now let $S$ be a proper subset of $\mathbb{Z}_n$. We aim to find a subset $T \subseteq \mathbb{Z}_n$ such that $S = T \cdot v$ for some word $v \in \{a, b\}^*$ and $|T| > |S|$. Since $n - |T| < n - |S|$, the induction assumption applies to the subset $T$ whence $T = \mathbb{Z}_n \cdot u$ for some word $u \in \{a, b\}^*$. Then $S = \mathbb{Z}_n \cdot uv$ is reachable as required.

Thus, fix a non-empty subset $S \subsetneqq \mathbb{Z}_n$. Since cosets of the trivial subgroup $H_0$ are singletons, $S$ is a union of cosets of $H_0$. On the other hand, since $H_\ell = \mathbb{Z}_n$, the only coset of $H_\ell$ strictly contains $S$, and so $S$ is not a union of cosets of $H_\ell$. Now choose $k \ge 1$ to be the maximal number for which $S$ is a union of cosets of the subgroup $H_{k-1}$. The subgroup $H_k$ already has a coset, say, $H_k \oplus t$ being neither contained in $S$ nor disjoint with $S$; in other words, $\varnothing \ne S \cap (H_k \oplus t) \subsetneqq H_k \oplus t$.

By Lemma 1.1, the coset $H_k \oplus t$ serves as the vertex set of a strongly connected component of the Cayley digraph $\mathrm{Cay}(\mathbb{Z}_n, D_k)$. Therefore, some edge of $\mathrm{Cay}(\mathbb{Z}_n, D_k)$ connects $(H_k \oplus t) \setminus S$ with $S \cap (H_k \oplus t)$ in this strongly connected component, that is, the source $q$ of this edge lies in $(H_k \oplus t) \setminus S$ while its target $p$ belongs to $S \cap (H_k \oplus t)$. Figure 2.3 illustrates this situation, the shaded area represents the subset $S$.

Let $p' = p \ominus q$; then $p' \in D_k$ by the definition of the Cayley digraph. By (2.3) there exists a word $w \in \{a, b\}^*$ such that $p' \in \mathrm{dupl}(w)$ and $\mathrm{excl}(w) \subseteq$

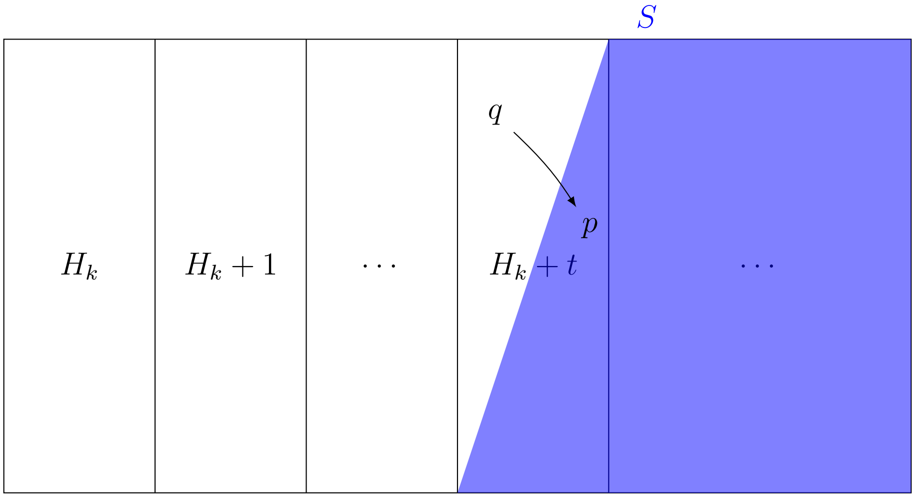


Figure 2.3: The position of the subset $S$ (shaded), and the edge $q \to p$ relative to the cosets of $H_k$.

$H_{k-1}$. Then

$$p = p' \oplus q = p' \cdot b^q \in \mathrm{dupl}(w) \cdot b^q = \mathrm{dupl}(wb^q)$$

and

$$\mathrm{excl}(wb^q) = \mathrm{excl}(w) \cdot b^q = \mathrm{excl}(w) \oplus q \subseteq H_{k-1} \oplus q.$$

From $p \in \mathrm{dupl}(wb^q)$ we conclude that there exist $p_1, p_2 \in \mathbb{Z}_n$ such that $p = p_1 \cdot wb^q = p_2 \cdot wb^q$. Since $S$ is a union of cosets of the subgroup $H_{k-1}$, the fact that $q \notin S$ implies that the whole coset $H_{k-1} \oplus q$ is disjoint with $S$, and the inclusion $\mathrm{excl}(wb^q) \subseteq H_{k-1} \oplus q$ ensures that $S$ is disjoint with $\mathrm{excl}(wb^q)$. A representation of this situation can be seen in Figure 2.4. Therefore, for every $s \in S \setminus \{p\}$, there exists a state $s' \in \mathbb{Z}_n$ such that $s' \cdot wb^q = s$. Now letting $T = \{p_1, p_2\} \cup \{s' \mid s \in S \setminus \{p\}\}$, we conclude that $S = T \cdot wb^q$ and $|T| = |S| + 1$.

□

For an illustration, return one last time to the automaton $\mathcal{E}'_{12}$ shown in Figure 2.2. We have seen that the subgroup $H_1(\mathcal{E}'_{12})$ consists of even residues modulo 12. Inspecting the word $ab^3a$ gives $\mathrm{excl}(ab^3a) = \{0, 8\} \subseteq H_1(\mathcal{E}'_{12})$ and $1 \in \mathrm{dupl}(ab^3a)$, whence $1 \in D_2(\mathcal{E}'_{12})$. Therefore the subgroup $H_2(\mathcal{E}'_{12})$ generated by $D_2(\mathcal{E}'_{12})$ is equal to $\mathbb{Z}_{12}$, and $\mathcal{E}'_{12}$ is a completely reachable automaton by Proposition 2.5.

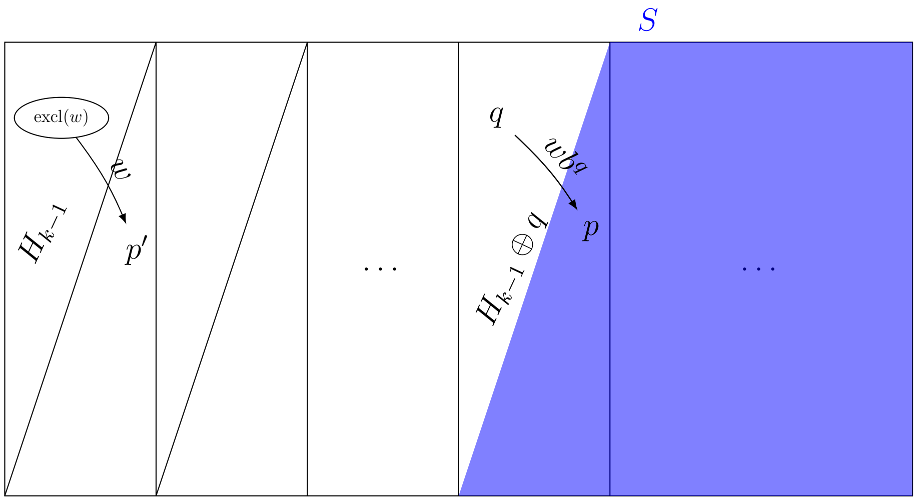


Figure 2.4: The position of the subset $S$ (shaded) with respect to the cosets of $H_{k-1}$ which are shown as triangles.

To illustrate the next level of the construction (2.3), consider the standardized automaton $\mathcal{E}_{48} = \langle \mathbb{Z}_{48}, \{a, b\}\rangle$ shown in Figure 2.5. We have replaced edges that should have been labeled $a$ and $b$ with solid and, resp., dashed edges and omitted all loops to lighten the picture. The action of $a$ in $\mathcal{E}_{48}$ is defined by $0 \cdot a = 24 \cdot a = 18$, $13 \cdot a = 14$, $14 \cdot a = 13$, $18 \cdot a = 24$, $30 \cdot a = 32$, $32 \cdot a = 30$, and $k \cdot a = k$ for all other $k \in \mathbb{Z}_{48}$.

One can calculate that $D_1(\mathcal{E}_{48}) = \{18, 24, 42\}$ whence the subgroup $H_1(\mathcal{E}_{48})$ consists of all residues divisible by 6. Computing $D_2(\mathcal{E}_{48})$, one sees that this set consists of even residues and contains 2 (due to the word $ab^{32}a$ that has $\mathrm{excl}(ab^{32}a) = \{0, 30\} \subseteq H_1(\mathcal{E}_{48})$ and $\mathrm{dupl}(ab^{32}a) = \{2, 18\}$). Hence the subgroup $H_2(\mathcal{E}_{48})$ consists of all even residues. Finally, the word $ab^{24}ab^{12}ab^{8}$ has $\{0, 8, 20\} \subseteq H_1(\mathcal{E}_{48})$ as its excluded set while its duplicate set contains 13. Hence $13 \in D_3(\mathcal{E}_{48})$ and the subgroup $H_3(\mathcal{E}_{48})$ coincides with $\mathbb{Z}_{48}$. We conclude that the automaton $\mathcal{E}_{48}$ is completely reachable by Proposition 2.5.

As mentioned, the subgroups of $(\mathbb{Z}_n, \oplus)$ ordered by inclusion correspond to the divisors of $n$ ordered by division. Hence, for any standardized automaton $\mathcal{A}$ with $n$ states the number of different subgroups of the form $H_k(\mathcal{A})$ is $O(\log n)$. Therefore, if the subgroup sequence $H_0(\mathcal{A}) \subseteq H_1(\mathcal{A}) \subseteq \cdots \subseteq H_k(\mathcal{A}) \subseteq \dots$ strictly grows at each step, then it reaches $\mathbb{Z}_n$ after at most $O(\log n)$ steps, and by Proposition 2.5 $\mathcal{A}$ is a completely reachable automaton. What happens if the sequence stabilizes earlier? Our next result answers this question.

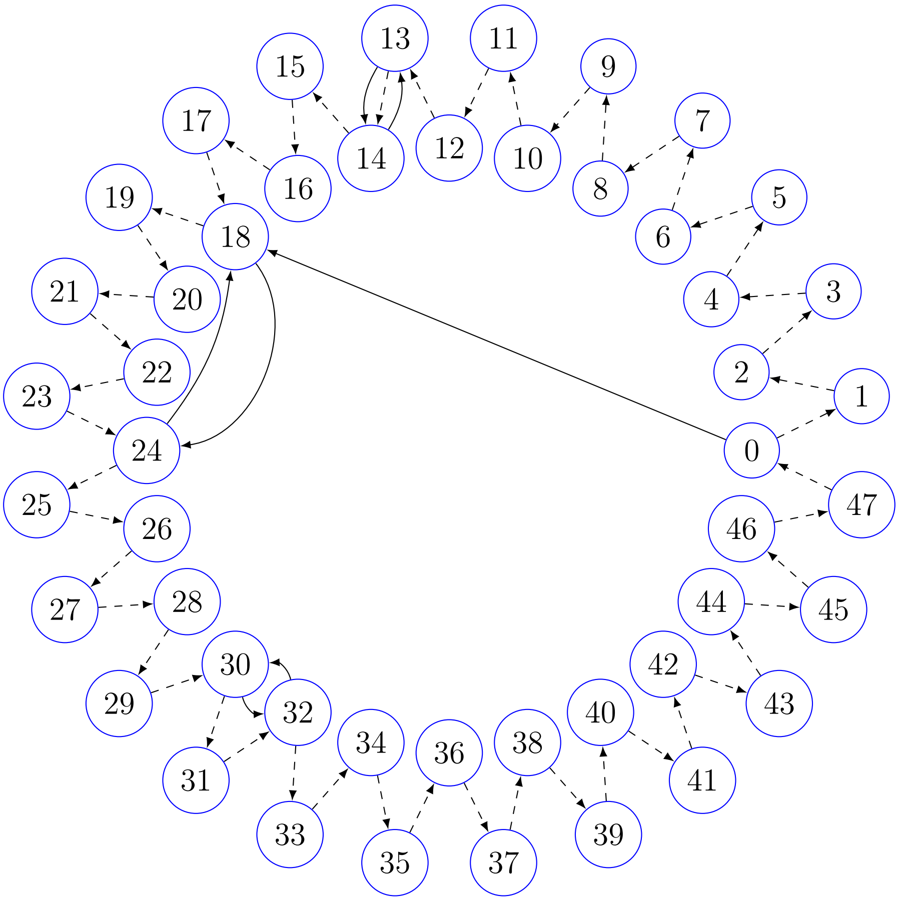


Figure 2.5: The automaton $\mathcal{E}_{48} = \langle \mathbb{Z}_{48}, \{a, b\} \rangle$ with $H_2(\mathcal{E}_{48}) \neq \mathbb{Z}_{48}$. Solid and dashed edges show the action of $a$ and, resp., $b$; loops are not shown

**Proposition 2.6.** *If for a standardized automaton $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\} \rangle$, there exists $\ell$ such that $H_\ell(\mathcal{A}) = H_{\ell+1}(\mathcal{A}) \subsetneqq \mathbb{Z}_n$, then $\mathcal{A}$ is not completely reachable.*

*Proof.* As in the proof of Proposition 2.5, we use $D_k$ and $H_k$ instead of $D_k(\mathcal{A})$ and, resp., $H_k(\mathcal{A})$ in our arguments.

It suffices to prove the following claim:

**Claim**: *the equality $H_\ell = H_{\ell+1}$ implies that the subgroup $H_\ell$ is $a$-invariant.*

Indeed, since $H_\ell \subsetneqq \mathbb{Z}_n$, we get a proper $a$-invariant subgroup, and Proposition 2.1 then shows that $\mathcal{A}$ is not completely reachable.

Technically, it is more convenient to show that if $H_\ell = H_{\ell+1}$, then $H_k \cdot a \subseteq H_\ell$ for every $k = 0, 1, \ldots, \ell$. We induct on $k$. The base $k = 0$ is clear since $H_0 = \{0\}$ and $0 \cdot a = \text{dupl}(a) \in D_1 \subseteq H_1 \subseteq H_\ell$.

Let $k < \ell$ and assume $H_k \cdot a \subseteq H_\ell$; we aim to verify that $p \cdot a \in H_\ell$ for every $p \in H_{k+1}$. Since the subgroup $H_{k+1}$ is generated by $D_{k+1}$ and contains $H_k$, we may choose a representation of $p$ as the sum

$$p = q \oplus d_1 \oplus \cdots \oplus d_m, \quad q \in H_k,\ d_1, \dots, d_m \in D_{k+1} \setminus H_k,$$

with the least number $m$ of summands from $D_{k+1} \setminus H_k$. We show that $p \cdot a \in H_\ell$ by induction on $m$. If $m = 0$, we have $p = q \in H_k$ and $p \cdot a \in H_\ell$ since $H_k \cdot a \subseteq H_\ell$.

If $m > 0$, we write $p$ as $p = d_1 \oplus s$ where $s = q \oplus d_2 \oplus \cdots \oplus d_m$. By (2.3), there exists a word $w \in \{a, b\}^*$ such that $d_1 \in \mathrm{dupl}(w)$, $0 \in \mathrm{excl}(w) \subseteq H_k$ and $|\mathrm{excl}(w)| \leq k + 1$. Consider the word $wb^s a$. We have

$$p \cdot a = (d_1 \oplus s) \cdot a = d_1 \cdot b^s a,$$

and the equality (1.2) gives $p \cdot a \in \mathrm{dupl}(wb^s a)$. From the equality (1.1), we get $\mathrm{excl}(wb^s a) = (\mathrm{excl}(w) \oplus s) \cdot a \cup \{0\}$ if $\mathrm{dupl}(a)a^{-1}$ is either contained in or disjoint with $\mathrm{excl}(w) \oplus s$, and

$$\mathrm{excl}(wb^s a) = \big((\mathrm{excl}(w) \oplus s) \setminus \mathrm{dupl}(a)a^{-1}\big) \cdot a \cup \{0\}$$

if $|\mathrm{dupl}(a)a^{-1} \cap (\mathrm{excl}(w) \oplus s)| = 1$. In any case, we have the inclusion

$$\mathrm{excl}(wb^s a) \subseteq (\mathrm{excl}(w) \oplus s) \cdot a \cup \{0\} \tag{2.4}$$

and the inequality

$$|\mathrm{excl}(wb^s a)| \leq |(\mathrm{excl}(w) \oplus s) \cdot a| + 1 \leq |\mathrm{excl}(w)| + 1 \leq (k+1) + 1 \leq \ell + 1. \tag{2.5}$$

For any $t \in \mathrm{excl}(w) \subseteq H_k$, the number of summands from $D_{k+1} \setminus H_k$ in the sum

$$t \oplus s = t \oplus q \oplus d_2 \oplus \cdots \oplus d_m$$

is less than $m$. By the induction assumption, we have $(t \oplus s) \cdot a \in H_\ell$. Hence, $(\mathrm{excl}(w) \oplus s) \cdot a \subseteq H_\ell$, and since 0 also lies in the subgroup $H_\ell$, we conclude from (2.4) that $\mathrm{excl}(wb^s a) \subseteq H_\ell$. From this and the inequality (2.5), we see that the word $wb^s a$ satisfies the conditions of the definition of $D_{\ell+1}$ (cf. (2.3)) whence every state in $\mathrm{dupl}(wb^s a)$ belongs to $D_{\ell+1}$. We have observed that $p \cdot a \in \mathrm{dupl}(wb^s a)$. Hence $p \cdot a \in D_{\ell+1} \subseteq H_{\ell+1}$. Since $H_\ell = H_{\ell+1}$, we have $p \cdot a \in H_\ell$, as required. □

Now we deduce a criterion for complete reachability of binary automata.

**Theorem 2.1.** *A binary automaton $\mathcal{A}$ with $n$ states is completely reachable if and only if either $n = 2$ and $\mathcal{A}$ is the flip-flop or one of the letters of $\mathcal{A}$ acts as a cyclic permutation of the state set, the other letter has defect* 1*, and in the standardized automaton $\langle \mathbb{Z}_n, \{a, b\} \rangle$ syntactically equivalent to $\mathcal{A}$, no proper subgroup of $(\mathbb{Z}_n, \oplus)$ is $a$-invariant.*

*Proof.* Necessity follows from the considerations made at the beginning of this chapter and Proposition 2.1.

For sufficiency, we can assume that $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\} \rangle$ is standardized. If no proper subgroup of $(\mathbb{Z}_n, \oplus)$ is $a$-invariant, then the claim from the proof of Proposition 2.6 implies that the sequence

$$H_0(\mathcal{A}) \subseteq H_1(\mathcal{A}) \subseteq \dots \subseteq H_k(\mathcal{A}) \subseteq \dots$$

strictly grows as long as the subgroup $H_k(\mathcal{A})$ remains proper. Hence, $H_\ell(\mathcal{A}) = \mathbb{Z}_n$ for some $\ell$ and $\mathcal{A}$ is a completely reachable automaton by Proposition 2.5. □

**Remark 2.1.** *The proof of Theorem 2.1 shows that only subgroups that contain $H_1(\mathcal{A})$ matter. Therefore, one can combine Theorem 1.1, Proposition 2.2 and Theorem 2.1 as follows:* a standardized automaton $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\} \rangle$ is completely reachable if and only if either $H_1(\mathcal{A}) = \mathbb{Z}_n$ or no proper subgroup of $(\mathbb{Z}_n, \oplus)$ containing the subgroup $H_1(\mathcal{A})$ is $a$-invariant.

For completeness sake, we state the main theorem for the non-standardized case highlighting the transformations for which the proper subgroups should not be invariant.

**Theorem 2.2.** *A binary automaton $\mathcal{A} = \langle Q, \{a, b\} \rangle$ with $n > 2$ states is completely reachable if and only if*

- *one of the letters of $\mathcal{A}$ acts as a cyclic permutation of the state set, the other letter has defect* 1*, $b$ and $a$ respectively;*
- *if* $\mathrm{coll}(a) = \{q_1, q_2\}$ *and* $0 \leq k_1, k_2 \leq n$ *are such that* $\mathrm{excl}(a) \cdot b^{k_1} = q_1$ *and* $\mathrm{excl}(a) \cdot b^{k_2} = q_2$*, then no proper subgroup of $(\mathbb{Z}_n, \oplus)$ is invariant either for the transformation $b^{k_1}a$ or $b^{k_2}a$.*

The condition of Theorem 2.1 can be verified in low polynomial time. We show how to do this in Algorithm 2. At first, given a binary automaton $\mathcal{A}$ with $n$ states, we first check if $n = 2$ and $\mathcal{A}$ is the flip-flop. If **yes**, $\mathcal{A}$ is completely reachable. If **not**, we check whether one of the letters of $\mathcal{A}$ acts as a cyclic permutation of the state set while the other letter has defect 1. This is done in line 3; in order to simplify we do not ask for the verification of this in the other way, but it is implicit. If **not**, $\mathcal{A}$ is not completely reachable. If **yes**, we pass to the standardized automaton $\langle \mathbb{Z}_n, \{a, b\} \rangle$ syntactically equivalent to $\mathcal{A}$. As a preprocessing, we compute and store the set $\{(k, k \cdot a) \mid k \in \mathbb{Z}_n\}$.

The rest of the algorithm can be stated in purely arithmetical terms. Call a positive integer $d$ a *nontrivial divisor* of $n$ if $d$ divides $n$ and $d \neq 1, n$.

We compute all nontrivial divisors of $n$ by checking through all integers $d = 2, \ldots, \lfloor\sqrt{n}\rfloor$: if such $d$ divides $n$, we store $d$ and $\frac{n}{d}$. If for some nontrivial divisor $d$ of $n$, all numbers $(td) \cdot a$ with $t = 0, 1, \ldots, \frac{n}{d} - 1$ are divisible by $d$, then $d$ generates a proper $a$-invariant subgroup in $(\mathbb{Z}_n, \oplus)$ and $\mathcal{A}$ is not completely reachable. If for every nontrivial divisor $d$ of $n$, there exists $t \in \{0, 1, \ldots, \frac{n}{d} - 1\}$ such that $(td) \cdot a$ is not divisible by $d$, then no proper subgroup of $(\mathbb{Z}_n, \oplus)$ is $a$-invariant and $\mathcal{A}$ is completely reachable.

**Algorithm 2** Decide Complete Reachability for Binary Automata

```
Input: A binary automaton $\mathcal{A} = \langle Q, \{a, b\}\rangle$ with $n \geq 2$ states.
if $n = 2$ and $\mathcal{A}$ is the flip-flop then
    return true
if not ($|Q \cdot a| = n - 1$ and
        $b$ acts as a cyclic permutation) then
    return false
Standardize $\mathcal{A}$.
for $d = 2, \ldots, \sqrt{n}$ do
    if $d|n$ then
        for $t = 0, \ldots, \frac{n}{d} - 1$ do
            DIVIDESALLIMAGES ← true
            if not $d|(td) \cdot a$ then
                DIVIDESALLIMAGES ← false
                break //There is no need to keep testing.
        if DIVIDESALLIMAGES then
            return false
        //We repeat the same process with $\frac{n}{d}$.
        for $t = 0, \ldots, d - 1$ do
            DIVIDESALLIMAGES ← true
            if not $\frac{n}{d}|(t\frac{n}{d}) \cdot a$ then
                DIVIDESALLIMAGES ← false
                break
        if DIVIDESALLIMAGES then
            return false
return true
```

To estimate the time complexity of the described procedure, observe that one has to check at most $\frac{n}{d}$ numbers for each nontrivial divisor $d$ of $n$. Clearly,

$$\sum_{\substack{1<d<n \\ d|n}} \frac{n}{d} = \sum_{\substack{1<d<n \\ d|n}} d = \sigma(n) - n - 1,$$

where $\sigma(n)$ stands for the sum of all divisors of $n$, a well-studied function in the theory of numbers; see, e.g., [22, Chapters XVI–XVIII]. It is known that $\limsup \frac{\sigma(n)}{n \log\log n} = e^{\gamma}$ where $\gamma$ is the Euler–Mascheroni constant [22, Theorem 323]; this implies that the number of checks in our procedure is $O(n \log\log n)$. The total complexity depends on the time spent for verifying the divisibility condition. If one uses the transdichotomous model [18], assuming constant time for division, the whole procedure can be implemented in $O(n \log\log n)$ time. One can speed up the above algorithm, using Remark 2.1, which implies that only the divisors $d > 1$ of the g.c.d. of $n$ and $0 \cdot a$ have to be checked. However, the improvement only reduces the constant behind the $O(\ )$ notation.

# Chapter 3

# Binary Completely Reachable Automata and Don's Conjecture

Once we have a characterization of completely reachable automata with two letters, the next natural problem to solve is to bound the length of the words that reach the subsets of states. We discussed before that it was Don who proposed a first conjecture on this issue in [15]. The bound he proposed was proved in the same article for binary completely reachable automata with a specific property. Let $\mathcal{A} = \langle Q, \{a, b\}\rangle$ be a binary automaton with $n > 2$ states and completely reachable. In this particular case $\mathcal{A}$ does not need to be standardized. Let $d > 1$ be the distance in the cycle determined by $b$ from $\mathrm{excl}(a)$ to $\mathrm{dupl}(a)$; this means that $\mathrm{excl}(a) \cdot b^d = \mathrm{dupl}(a)$. In [15, Proposition 15] Don proved that if $d$ and $n$ are coprime then for every subset of size $k \geq 1$ there is a word of length $n(n-k)$ that reaches it.

Although Don's conjecture have been proved false in the general case, a work to be done is to find classes of automata where the suggested bound is true. Ferens and Szykuła, in [17], proved a not so weaker bound for all completely reachable automata, the double of the bound proposed by Don. Since the original suggested bound came from a kind of binary automata, it would make sense to try to generalize the result to all binary automata. But this is not the case. In [39] Yinfeng Zhu presents a series of binary automata for which Don's proposed bound does not hold. For every even $n \geq 12$, let $\mathscr{Z}_n = \langle \mathbb{Z}_n, \{a, b\}\rangle$ be a binary automaton such that the letter $a$, of defect 1, fixes all states except for $0, 1, \frac{n}{2}, n-4, n-3, n-2$ and $n-1$ and acts on these states as shown in Figure 3.1.

Zhu proves that the shortest word to reach the subset $\mathbb{Z}_n \setminus \{\frac{n}{2} - 1, n-1\}$ has a length of at least $\frac{5}{2}n - 3$.

Note that this series of automata are not standardized. After standardization the obtained series comply with Don's bound. At the moment of

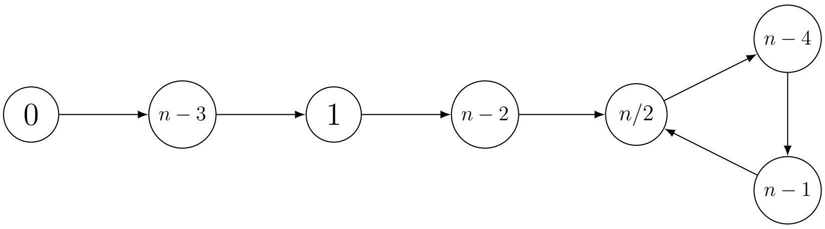


Figure 3.1: The action of the letter $a$ in Zhu's series of counter examples.

writing Don's conjecture is open for binary completely reachable automata in the standardized form. Nevertheless, some partial results have been obtained. For example, Don [15, Proposition 15] proved the following proposition, stated using the concepts used in this work for simplicity's sake:

**Proposition 3.1.** *Let $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\} \rangle$ be a binary standardized automaton such that $b$ is the cyclical permutation of $\mathbb{Z}_n$ and the letter $a$ has defect 1. If $d = \mathrm{dupl}(w)$ is coprime with $n$, then every subset of $\mathbb{Z}_n$ of size $k > 1$ can be reached by a word of length at most $n(n-k)$.*

Denote the state $0 \cdot a$ by $d$ and let $r$ stand for the state such that $r \neq 0$ and $r \cdot a = d$. The letter $a$ acts as a permutation on the set $\{1, \dots, n-1\}$(Lemma 2.3). Therefore, acting by a suitable power of the letter $a$ at the state $d$, one gets the state $r$, that is, $r = d \cdot a^{\ell-1}$ for some positive integer $\ell$. Let $\ell$ be the least positive integer with this property, and for each $s = 0, 1, \dots, \ell - 1$, let $d_s := d \cdot a^s$ so that $d_0 = d$, $d_{\ell-1} = r$, and all states $d_0, d_1, \dots, d_{\ell-1}$ are distinct. We denote the set $\{d_0, d_1, \dots, d_{\ell-1}\}$ by $\mathrm{orb}(d)$ and call it the *orbit* of the automaton $\langle \mathbb{Z}_n, \{a, b\} \rangle$. The subgroup of $(\mathbb{Z}_n, \oplus)$ generated by $\mathrm{orb}(d)$ is called the *orbit subgroup* of the automaton.

The main result of this chapter is the following:

**Theorem 3.1.** *Every standardized automaton $\langle \mathbb{Z}_n, \{a, b\} \rangle$ whose orbit subgroup coincides with the group $(\mathbb{Z}_n, \oplus)$ fulfils Don's conjecture.*

Theorem 3.1 generalizes Don's result([15, Proposition 15]) for standardized automata.

## 3.1 Expandable subsets

The two results collected in this section are basically known as their versions have been scattered over the literature; see, e.g., [15, 17, 8]. We believe it is more convenient for the reader to see direct arguments rather than follow references to various sources where similar ideas might have appeared under different terminology and notation. Therefore, we have included complete proofs without claiming any originality.

Let $\mathcal{A} = \langle Q, \Sigma \rangle$ be an automaton. We say that a word $w \in \Sigma^*$ *expands* a proper non-empty subset $S \subset Q$ if there exists a subset $P \subseteq Q$ such that $|P| > |S|$ and $P \cdot w = S$. The following easy observation connects this notion with the concepts of excluded and duplicated sets.

**Lemma 3.1.** *Let $\mathcal{A} = \langle Q, \Sigma \rangle$ be an automaton. A word $w \in \Sigma^*$ expands a proper non-empty subset $S \subset Q$ if and only if* $\mathrm{excl}(w) \cap S = \emptyset$ *while* $\mathrm{dupl}(w) \cap S \neq \emptyset$.

*Proof.* For the 'only if' part, let $P$ be a subset of $Q$ with $|P| > |S|$ and $P \cdot w = S$. Since $S = P \cdot w \subseteq Q \cdot w$, we get $S \cap (Q \backslash Q \cdot w) = \emptyset$, that is, $\mathrm{excl}(w) \cap S = \emptyset$. Since $|P| > |P \cdot w|$, there exist some $p, p' \in P$ such that $p \neq p'$ but $p \cdot w$ coincides with $p' \cdot w$. Then $p \cdot w \in \mathrm{dupl}(w) \cap S$ whence $\mathrm{dupl}(w) \cap S \neq \emptyset$.

Conversely, for the 'if' part, let $w \in \Sigma^*$ be a word with $\mathrm{excl}(w) \cap S = \emptyset$ and $\mathrm{dupl}(w) \cap S \neq \emptyset$. Since $\mathrm{excl}(w) = Q \backslash Q \cdot w$ is disjoint from $S$, we have $S \subseteq Q \cdot w$. Hence for every state $s \in S$, its preimage $sw^{-1} := \{q \in Q \mid q \cdot w = s\}$ is non-empty. Let $P := \bigcup\limits_{s \in S} sw^{-1}$. Then $P \cdot w = S$ and $|P| > |S|$ since the subsets $sw^{-1}$ are disjoint and for each $p \in \mathrm{dupl}(w) \cap S$, the set $pw^{-1}$ is non-singleton. □

Given an automaton $\mathcal{A} = \langle Q, \Sigma \rangle$ with $|Q| = n$, a proper non-empty subset of $Q$ is said to be *$n$-expandable* if it can be expanded by a word of length at most $n$.

**Lemma 3.2.** *If in an automaton with $n$ states, every proper non-empty subset is $n$-expandable, then every subset with $k > 0$ states is reachable by a word of length $\leq n(n-k)$.*

*Proof.* Let an automaton $\mathcal{A} = \langle Q, \Sigma \rangle$ satisfy the premise so that $|Q| = n$. We prove that for any $k$ with $0 < k \leq n$, every subset $S \subseteq Q$ with $k$ states is reachable by a word of length $\leq n(n-k)$ by induction on $n-k$.

If $n-k = 0$, then $S = Q$ and the claim holds since $Q$ is reachable by the empty word whose length is 0.

Now let $n-k > 0$ so that $S$ is a proper subset of $Q$. Then $S$ is $n$-expandable so that there exist a word $w \in \Sigma^*$ of length at most $n$ and a subset $P \subseteq Q$ such that $|P| > |S|$ and $P \cdot w = S$. Since $|P| > |S| = k$, we have $n - |P| < n - k$, and the induction assumption applies to the subset $P$. Hence, $P = Q \cdot v$ for some word $v \in \Sigma^*$ of length $\leq n(n-k-1)$. Then

$S = P \cdot w = (Q \cdot v) \cdot w = Q \cdot vw$ and the length of the word $vw$ does not exceed $n(n-k-1)+n = n(n-k)$ as required. □

Lemmas 3.1 and 3.2 imply that an automaton $\mathcal{A} = \langle Q, \Sigma \rangle$ satisfies Don's conjecture whenever for each proper non-empty subset $S \subset Q$, one can find a word $w$ of length at most $|Q|$ with $\mathrm{excl}(w) \cap S = \emptyset$ and $\mathrm{dupl}(w) \cap S \neq \emptyset$.

One caveat seems to be in order: our notion of expandability should not be confused with extensibility, a similar but different concept widely used in the theory of synchronizable automata. We discuss this in some detail later in this chapter.

## 3.2 The restricted orbit digraph

In this part recall the definition of the Cayley graph of a group presented in Definition 1.2, together with Lemma 1.1 that describes the structure of its strongly connected components. The *orbit digraph* $\mathcal{O}(\mathcal{A})$ of a standardized automaton $\mathcal{A} = \langle \mathbb{Z}_n, \{a,b\} \rangle$ is the Cayley digraph $\mathrm{Cay}(\mathbb{Z}_n, \mathrm{orb}(d))$. Denote the orbit subgroup of $\mathcal{A}$ by $H_0$. Thus, each edge of $\mathcal{O}(\mathcal{A})$ is of the form $q \to q \oplus d_s$, where $q \in \mathbb{Z}_n$ and $d_s \in \mathrm{orb}(d)$, and the strongly connected components of $\mathcal{O}(\mathcal{A})$ have the cosets $q \oplus H_0$, $q \in \mathbb{Z}_n$, as their vertex sets.

The edge $q \to q \oplus d_s$ is called *long* if $q + s \geq n$ and *short* otherwise. The *restricted orbit digraph* $\mathcal{R}(\mathcal{A})$ is the spanning subgraph of the orbit digraph obtained by removing all long edges from the latter digraph. We aim to show that what was connected in the orbit digraph remains so in the restricted orbit digraph. Before that let us state a technical lemma that aid us in the proof of the proposition

**Lemma 3.3.** *Let $d_0, d_1, \ldots, d_{s-1}$ be distinct positive integers less than $n$. Then the greatest common divisor of $d_0, d_1, \ldots, d_{s-1}, n$ does not exceed $\frac{n}{s+1}$.*

*Proof.* Let $c_0, c_1, \ldots, c_{s-1}$ be the numbers $d_0, d_1, \ldots, d_{s-1}$ arranged in the ascending order. Denoting the greatest common divisor of $d_0, d_1, \ldots, d_{s-1}, n$ by $g$, we get $c_0 \geq g$, $c_1 \geq 2g$, …, $c_{s-1} \geq sg$, and finally, $n \geq (s+1)g$ since each of the numbers $c_0, c_1, \ldots, c_{s-1}, n$ is a multiple of $g$ and all these numbers are distinct. Hence, $g \leq \frac{n}{s+1}$. □

**Proposition 3.2.** *Let $\mathcal{A} = \langle \mathbb{Z}_n, \{a,b\} \rangle$ be a standardized automaton, and $H_0$ the subgroup of $(\mathbb{Z}_n, \oplus)$ generated by* $\mathrm{orb}(d)$. *The strongly connected components of the restricted orbit digraph $\mathcal{R}(\mathcal{A})$ have the cosets of the subgroup $H_0$ as their vertex sets.*

*Proof.* For a state $p \in \mathbb{Z}_n$, denote by $\langle p \rangle$ the subgroup of $(\mathbb{Z}_n, \oplus)$ generated by $p$. Let $\operatorname{orb}(d) = \{d_0, d_1, \ldots, d_{\ell-1}\}$, and for each $s = 0, 1, \ldots, \ell - 1$, let $g_s$ stand for the greatest common divisor of the numbers $d_0, d_1, \ldots, d_s, n$. We then have that $g_0$ is a multiple of $g_1$, which is a multiple of $g_2$, and so on.

Inducting on $s$, we will establish the following:

> **Claim:** *For each $s = 0, 1, \ldots, \ell - 1$, the restricted orbit digraph $\mathcal{R}(\mathcal{A})$ has a spanning subgraph $\Gamma^{(s)}$ whose strongly connected components have the cosets of the subgroup $\langle g_s \rangle$ as their vertex sets.*

*Proof of the claim.* For $s = 0$, consider the following $n$ edges in $\mathcal{O}(\mathcal{A})$:

$$0 \to d_0,\ 1 \to d_0 \oplus 1,\ \ldots,\ n - 1 \to d_0 \oplus (n - 1).$$

They all are short and easily seen to form $g_0$ directed cycles whose vertex sets are the cosets of the subgroup $\langle g_0 \rangle$. Thus, the spanning subgraph with these $n$ edges can be taken as $\Gamma^{(0)}$.

Now let $s > 0$. Construct a new spanning subgraph $\Gamma$ of the graph $\mathcal{O}(\mathcal{A})$ by adding to the digraph $\Gamma^{(s-1)}$ the following $g_{s-1}$ edges:

$$0 \to d_s,\ \ 1 \to d_s \oplus 1,\ \ldots,\ g_{s-1} - 1 \to d_s \oplus (g_{s-1} - 1).$$

First, we verify that each of these edges is short. For this, it suffices to show that $s + (g_{s-1} - 1) < n$, that is, $g_{s-1} + s \leq n$. We can conclude this with the aid of Lemma 3.3. Indeed,

$$\begin{aligned}
n - g_{s-1} - s &\geq n - \frac{n}{s+1} - s && \text{by Lemma 3.3} \\
&= n\left(1 - \frac{1}{s+1}\right) - s \\
&\geq (s+1)\left(1 - \frac{1}{s+1}\right) - s && \text{since } n > \ell \geq s \\
&= (s+1) - 1 - s = 0.
\end{aligned}$$

Back to the construction of the spanning subgraph $\Gamma$, we have to analyze its strongly connected components.

It readily follows from the definition of the greatest common divisor that $g_s$ is the greatest common divisor of $g_{s-1}$ and $d_s$. Hence $g_{s-1} = mg_s$ and $d_s = kg_s$ for some coprime $m$ and $k$. The subgroup $\langle g_s \rangle$ is equal to the union of the $m$ cosets of the subgroup $\langle g_{s-1} \rangle$ that are contained in $\langle g_s \rangle$; these $m$ cosets are

$$\langle g_{s-1} \rangle,\ \ g_s \oplus \langle g_{s-1} \rangle,\ \ldots,\ (m-1)g_s \oplus \langle g_{s-1} \rangle. \tag{3.1}$$

We have $d_s \in \overline{k}g_s \oplus \langle g_{s-1}\rangle$, where $\overline{k}$ is the residue of $k$ modulo $m$. Therefore, in the subgraph $\Gamma$, the newly added edge $0 \to d_s$ connects the strongly connected components $\langle g_{s-1}\rangle$ and $\overline{k}g_s \oplus \langle g_{s-1}\rangle$ of the subgraph $\Gamma^{(s-1)}$. In the same way, the edge $\overline{k}g_s \to d_s \oplus \overline{k}g_s$ connects the strongly connected components $\overline{k}g_s \oplus \langle g_{s-1}\rangle$ and $\overline{2k} \oplus \langle g_{s-1}\rangle$, where $\overline{2k}$ is the residue of $2k$ modulo $m$, etc. Since $m$ and $k$ are coprime, the $m$ edges

$$0 \to d_s,\ \overline{k}g_s \to d_s \oplus \overline{k}g_s,\ \overline{2k}g_s \to d_s \oplus \overline{2k}g_s,\ \ldots,\ \overline{(m-1)k}g_s \to d_s \oplus \overline{(m-1)k}g_s$$

cyclically connect all $m$ cosets in (3.1). By the induction assumption, each of these cosets is the vertex set of a strongly connected component of the digraph $\Gamma^{(s-1)}$. Hence, all states in the subgroup $\langle g_s\rangle$ are mutually reachable in the digraph $\Gamma$.

In the same way, for each $i = 1, \ldots, g_s - 1$, the $m$ edges

$$i \to d_s \oplus i,\ \overline{k}g_s \oplus i \to d_s \oplus \overline{k}g_s \oplus i,\ \overline{2k}g_s \oplus i \to d_s \oplus \overline{2k}g_s \oplus i,\ \ldots,$$
$$\overline{(m-1)k}g_s \oplus i \to d_s \oplus \overline{(m-1)k}g_s \oplus i$$

cyclically connect the $m$ cosets

$$i \oplus \langle g_{s-1}\rangle,\ i \oplus g_s \oplus \langle g_{s-1}\rangle,\ \ldots,\ i \oplus (m-1)g_s \oplus \langle g_{s-1}\rangle.$$

As above, using the induction assumption, we conclude that all states in the coset $i \oplus \langle g_s\rangle$ also are mutually reachable in the digraph $\Gamma$. Since no more edges were added when constructing the graph $\Gamma$, the $g_s$ cosets $i \oplus \langle g_s\rangle$ with $i = 0, 1, \ldots, g_s - 1$ form the vertex sets of the strongly connected components of $\Gamma$.

We have verified that the spanning subgraph $\Gamma$ fulfils all requirements we need, and thus, can be taken as $\Gamma^{(s)}$. This completes the proof of the inductive step, and hence, the proof of the claim. □

**Remark 3.1.** *It may happen that $g_{s-1}$ divides $d_s$, in which case $g_{s-1} = g_s$. In this situation the above construction of the spanning subgraph $\Gamma^{(s)}$ still works fine (with $m = 1$), because each newly added edge connects vertices within a strongly connected component of $\Gamma^{(s-1)}$. Thus, while having more edges, the graph $\Gamma^{(s)}$ has the same strongly connected components as $\Gamma^{(s-1)}$.*

The proof of Proposition 3.2 is now immediate since the subgroup $\langle g_{\ell-1}\rangle$ coincides with the subgroup $H_0$. □

For the rest of this chapter we use the following notation: Let $\mathbb{Z}_n$ be the cyclic group of $n$ elements, and $1 \leq m \leq n$ an arbitrary element of the group. The cyclic subgroup of $\mathbb{Z}_n$ generated by $m$, i.e., $\{0, m, 2m, \ldots\}$, is denoted by $m\mathbb{Z}_n$. As expected, for any element $1 \leq q \leq n$, the coset of $m\mathbb{Z}_n$ who contains $q$ is denoted by $q \oplus m\mathbb{Z}_n$.

For an illustration, we trace the inductive construction in the proof of Proposition 3.2 on the 48-state automaton $\mathcal{E}'_{48} = \langle \mathbb{Z}_{48}, \{a, b\} \rangle$ shown in Figure 3.2 below. In $\mathcal{E}'_{48}$, we have $d = 24$, and the orbit of $\mathcal{E}'_{48}$ consists of $d = d_0$ and $r = d_1 = 18$ so that $\ell = 2$.

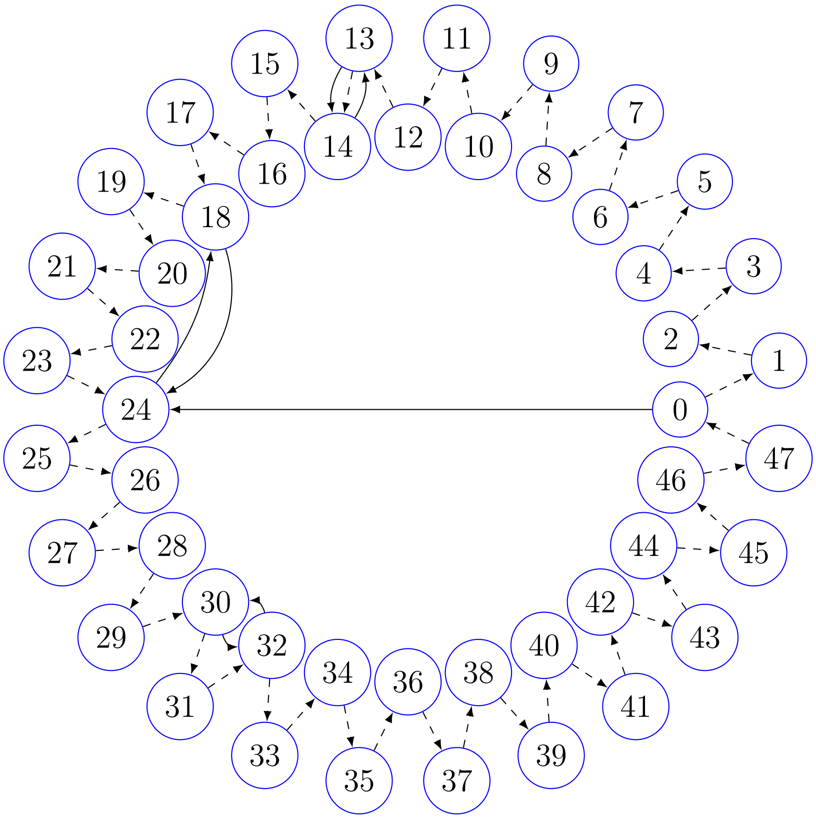
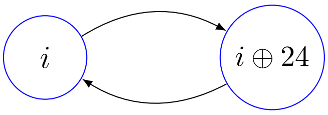


Figure 3.2: The automaton $\mathcal{E}'_{48} = \langle \mathbb{Z}_{48}, \{a, b\} \rangle$. Solid and dashed edges show the action of $a$ and, resp., $b$; if $a$ fixes a state, the corresponding loop is omitted to improve readability.

We have $g_0 = d_0 = 24$. Therefore, we begin the construction with the spanning subgraph $\Gamma^{(0)}$ that consists of the 24 directed cycles



with $i = 0, 1, \ldots, 23$, having the 24 cosets of the 2-element subgroup $(24\mathbb{Z}_{48}, \oplus)$ as the vertex sets. All edges in $\Gamma^{(0)}$ are short.

To get the next spanning subgraph $\Gamma^{(1)}$, we add to $\Gamma^{(0)}$ the following 24 edges:

$$0 \to 18,\ 1 \to 19,\ 2 \to 20,\ \ldots,\ 23 \to 41.$$

which are all short. The greatest common divisor $g_1$ of the numbers $d_0 = 24$, $d_1 = 18$ and 48 is 6. We have $m = \dfrac{g_0}{g_1} = 4$ and $k = \dfrac{d_1}{g_1} = 3$. The 8-element subgroup $\langle g_s \rangle = (6\mathbb{Z}_{48}, \oplus)$ is the union of the following four cosets of the group $(24\mathbb{Z}_{48}, \oplus)$:

$$24\mathbb{Z}_{48}, \quad 6 \oplus 24\mathbb{Z}_{48}, \quad 12 \oplus 24\mathbb{Z}_{48}, \quad 18 \oplus 24\mathbb{Z}_{48}.$$

The newly added edges $0 \to 18$, $18 \to 36$, $6 \to 24$, $12 \to 30$ cyclically connect these four cosets, producing a strongly connected component of $\Gamma^{(1)}$ as shown in Figure 3.3. The three other strongly connected components of the digraph $\Gamma^{(1)}$ are constructed in the same way.

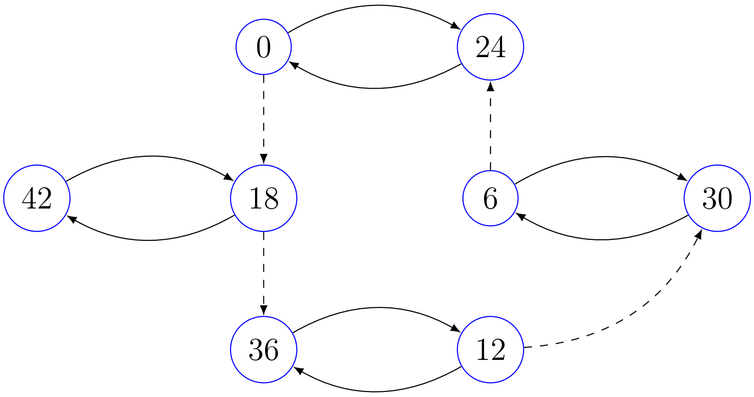


Figure 3.3: One of the strongly connected components of the digraph $\Gamma^{(1)}$ constructed for the automaton $\mathcal{E}'_{48}$ from Figure 2.5. The solid edges are inherited from $\Gamma^{(0)}$; the dashed edges are newly added.

As an application of Proposition 3.2, we infer that certain proper non-empty subsets in standardized automata are $n$-expandable.

**Proposition 3.3.** *Let $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\} \rangle$ be a standardized automaton and $H_0$ its orbit subgroup. Every non-empty subset of $\mathbb{Z}_n$ which is not a union of $H_0$-cosets is $n$-expandable.*

*Proof.* If a non-empty subset $S$ of $\mathbb{Z}_n$ is not a union of $H_0$-cosets, then there exists a coset $C$ which is neither contained in $S$ nor disjoint from $S$. Then there exist a state $p \in C\backslash S$ and a state $p' \in C \cap S$. Proposition 3.2 implies that any two states in $C$ are connected in the restricted orbit digraph $\mathcal{R}(\mathcal{A})$; in particular, there is a sequence $p = p_0, p_1, \dots, p_t = p'$ of states in $C$ such that $p_{i-1} \to p_i$ is an edge in $\mathcal{R}(\mathcal{A})$ for each $i = 1, \dots, t$. If $j$ is the maximal index such that $p_j \in C\backslash S$, then $j < t$ and $p_{j+1} \in C \cap S$. Renaming $p_j$ and $p_{j+1}$ to $q$ and $q'$, respectively, we conclude that the edge $q \to q'$ of $\mathcal{R}(\mathcal{A})$ is such that $q \notin S$ and $q' \in S$.

Let $\operatorname{orb}(d) = \{d_0, d_1, \dots, d_{\ell-1}\}$. By the construction of the restricted orbit digraph, $q \to q'$ being its edge means that $q \to q'$ is a short edge in the orbit digraph $\mathcal{O}(\mathcal{A})$. Unfolding the definitions of $\mathcal{O}(\mathcal{A})$ and of being short, we see that $q' = q \oplus d_s$ for some $s \in \{0, 1, \dots, \ell - 1\}$ and $q + s < n$. Now consider the word $a^{s+1}b^q$ of length $q + s + 1 \leq n$. Since

$$\left.\begin{array}{l} 0 \cdot a^{s+1}b^q \\ r \cdot a^{s+1}b^q \end{array}\right\} = d \cdot a^s b^q = d_s \cdot b^q = d_s \oplus q = q',$$

the duplicate set of $a^{s+1}b^q$ contains $q'$. On the other hand, the only excluded state of $a^{s+1}$ is 0 whence $\operatorname{excl}(a^{s+1}b^q) = \{q\}$. Thus, we have $\operatorname{excl}(a^{s+1}b^q) \cap S = \emptyset$ while $\operatorname{dupl}(a^{s+1}b^q) \cap S \neq \emptyset$. Lemma 3.1 then implies that the word $a^{s+1}b^q$ expands $S$. Since the length of this word does not exceed $n$, the subset $S$ is $n$-expandable. □

Now we can easily deduce Theorem 3.1.

*Proof of Theorem 3.1.* Let $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\} \rangle$ be a standardized automaton whose orbit subgroup $H_0$ coincides with $(\mathbb{Z}_n, \oplus)$. Then no non-empty proper subset of $\mathbb{Z}_n$ can be a union of $H_0$-cosets, whence each non-empty proper subset of $\mathbb{Z}_n$ is $n$-expandable by Proposition 3.3. Now Lemma 3.2 implies that every subset with $k > 0$ states is reachable by a word of length $\leq n(n - k)$. Thus, the automaton $\mathcal{A}$ fulfills Don's conjecture. □

## 3.3 Further discussion

Analysing the above proof of Theorem 3.1, we see that a stronger statement has actually been proved: in every standardized automaton $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\} \rangle$ whose orbit subgroup $H_0$ coincides with $(\mathbb{Z}_n, \oplus)$, each subset with $k > 0$ states is reachable by a product of $n - k$ words of the form $a^{i+1}b^j$ where $0 \leq i, j \leq n-1$ and $i + j \leq n$. Each word of the form $a^{i+1}b^j$ with $0 \leq i, j \leq n-1$ has a unique excluded state (namely, $\operatorname{excl}(a^{i+1}b^j) = \{j\}$). For any word $w$, its *defect* is defined as the cardinality of the set $\operatorname{excl}(w)$. Thus, under the premise of Theorem 3.1, each subset with $k > 0$ states is reachable by a product of $n - k$ words of defect 1 and length $\leq n$.

Recall from Section 1.2 that an automaton is perfectly reachable if each subset with $k > 0$ states is reachable in $\mathcal{A}$ by a product of $|Q| - k$ words of defect 1.

Turning back to binary automata, observe that for any standardized automaton $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\} \rangle$, its orbit graph is a spanning subgraph of the Rystsov graph $\Gamma_1(\mathcal{A})$. Indeed, the edges of the orbit digraph are of the form $q \to q \oplus d_s$, where $q \in \mathbb{Z}_n$ and $d_s \in \mathrm{orb}(d)$. The word $w_{s,q} := a^s b^q$ has defect 1 and $\mathrm{excl}(w_{s,q}) = q$ while $\mathrm{dupl}(w_{s,q}) = q \oplus d_s$. Hence the edge $q \to q \oplus d_s$ occurs in the Rystsov digraph as the edge forced by $w_{s,q}$. By Lemma 1.1, automata satisfying the premise of Theorem 3.1 are precisely standardized automata with strongly connected orbit graphs. Hence, such automata are perfectly reachable by Proposition 1.1.

Attempting to extend our approach to arbitrary, perfectly reachable standardized automata, one may define restricted versions of Rystsov graphs parallel to restricted orbit graphs of Subsection 3.2. Namely, the *restricted Rystsov digraph* of $\mathcal{A} = \langle \mathbb{Z}_n, \{a, b\} \rangle$ is the spanning subgraph of $\Gamma_1(\mathcal{A})$ in which one retains only edges forced by words of length at most $n$. In order to transfer the arguments of Subsection 3.2 to perfectly reachable standardized automata, one needs to establish an analog of Proposition 3.2, that is, to show that what was connected in $\Gamma_1(\mathcal{A})$ remains so in the restricted Rystsov graph. However, as the following example demonstrates, this is not true in general.

**Example 3.1.** Consider the standardized automaton $\mathcal{E}_{12} = \langle \mathbb{Z}_{12}, \{a, b\} \rangle$ shown in Figure 3.4. Observe that $0 \cdot a = 10 = 10 \cdot a$ in $\mathcal{E}_{12}$ so that for this automaton, both parameters $d$ and $r$ are equal to 10 and the orbit $\mathrm{orb}(d)$ reduces to the singleton $\{10\}$. Therefore, the orbit digraph of $\mathcal{E}_{12}$ has two strongly connected components; they have as the vertex sets the subgroup $(2\mathbb{Z}_{12}, \oplus)$ of all even residues modulo 12 and its coset $1 \oplus 2\mathbb{Z}_{12}$ consisting of all odd residues. In contrast, the Rystsov digraph $\Gamma_1(\mathcal{E}_{12})$ is strongly connected. This claim can be verified by either brute force successive checking through all words of defect 1 or invoking Propositions 2 and 3 of [11] that characterize $\Gamma_1(\mathcal{E}_{12})$ as the Cayley digraph $\mathrm{Cay}(\mathbb{Z}_{12}, D)$ where

$$D = \{d \cdot v \mid v \in \{a, b^r a\}^*\} = \{10 \cdot v \mid v \in \{a, b^{10} a\}^*\}.$$

Going either way, one eventually finds the word $(ab^{10})^4 a$ of length 45 that has defect 1 and forces the edge $0 \to 1$ of $\Gamma_1(\mathcal{E}_{12})$. The word $(ab^{10})^4 ab$ also has defect 1 and forces the edge $1 \to 2$ in $\Gamma_1(\mathcal{E}_{12})$. The two edges $0 \to 1$ and $1 \to 2$ connect the strongly connected components of the orbit digraph of

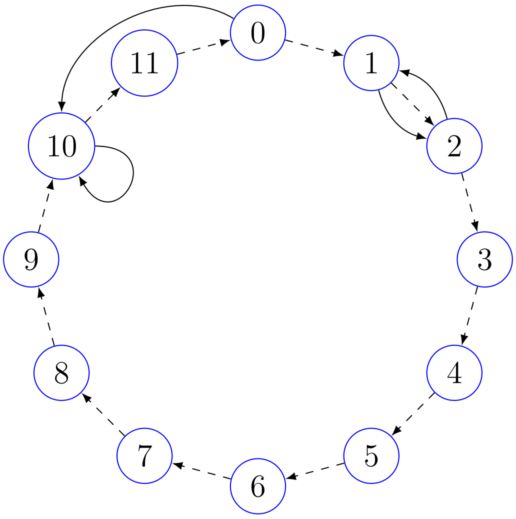


Figure 3.4: The automaton $\mathcal{E}_{12} = \langle \mathbb{Z}_{12}, \{a, b\} \rangle$. Solid and dashed edges show the action of $a$ and, resp., $b$.

$\mathcal{E}_{12}$, and thus, ensure strong connectivity of $\Gamma_1(\mathcal{E}_{12})$. By Proposition 1.1 the automaton $\mathcal{E}_{12}$ is perfectly reachable.

Computing all words of defect 1 and length at most 12, one gets the restricted Rystsov digraph of $\mathcal{E}_{12}$ shown in Figure 3.5. This graph is not strongly connected.

Thus, the method we used to prove Theorem 3.1 cannot be directly extended to show that Don's conjecture holds for perfectly reachable standardized automata. Of course, this does not disprove the conjecture. In particular, the automaton $\mathcal{E}_{12}$ is not a counterexample to Don's conjecture. By Lemma 3.2, to justify the latter claim, it suffices to show that every proper non-empty subset of $\mathbb{Z}_{12}$ is 12-expandable in $\mathcal{E}_{12}$. Proposition 3.3 ensures this for all subsets except for $2\mathbb{Z}_{12}$ and $1 \oplus 2\mathbb{Z}_{12}$, and one easily verifies that the word $aba$ expands $1 \oplus 2\mathbb{Z}_{12}$ while the word $(ab)^2$ does the job for $2\mathbb{Z}_{12}$. (The words $aba$ and $(ab)^2$ have defect 2, and therefore, they do not show up when the restricted Rystsov graph is constructed.)

The previous example shows that expanding a subset with a short word can become possible if using words of defect greater than 1 is allowed. It is natural to ask whether this trick solves the issue for all perfectly reachable standardized automata. If so, then every proper non-empty subset in each $n$-state perfectly reachable standardized automaton would be $n$-expandable and Don's conjecture for such automata would follow by Lemma 3.2. How-

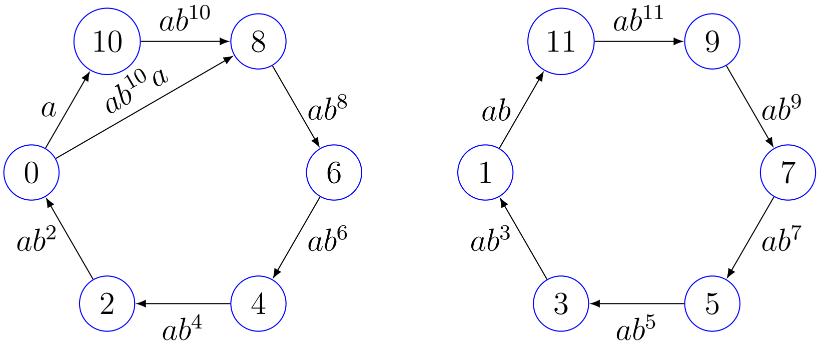


Figure 3.5: The restricted Rystsov digraph of the automaton $\mathcal{E}_{12}$. Each edge is labeled by the shortest word of defect 1 forcing it.

ever, our next example exhibits a 21-state perfectly reachable standardized automaton with a subset that fails to be 21-expandable.

**Example 3.2.** Consider the standardized automaton $\mathcal{E}_{21} = \langle \mathbb{Z}_{21}, \{a, b\} \rangle$ shown in Figure 3.6 where all loops labeled $a$ are omitted to lighten the picture. Observe that $0 \cdot a = 14 = 14 \cdot a$ in $\mathcal{E}_{21}$. Thus, for $\mathcal{E}_{21}$, both parameters $d$ and $r$ are equal to 14 and the orbit $\operatorname{orb}(d)$ reduces to the singleton $\{14\}$. The orbit graph of $\mathcal{E}_{21}$ has seven strongly connected components whose the vertex sets are the subgroup $(7\mathbb{Z}_{21}, \oplus)$ generated by 14 and its cosets. The word $ab^{14}a$ has defect 1, $\operatorname{excl}(ab^{14}a) = \{0\}$ and $\operatorname{dupl}(ab^{14}a) = \{18\}$. Hence, the Rystsov digraph $\Gamma_1(\mathcal{E}_{21})$ has the edge $0 \to 18$. Multiplying $ab^{14}a$ on the right by $b, b^2, b^3, b^4, b^4, b^6$, we get words of defect 1 that force the edges

$$1 \to 19,\ 2 \to 20,\ 3 \to 0,\ 4 \to 1,\ 5 \to 2,\ 6 \to 3$$

in $\Gamma_1(\mathcal{E}_{21})$. These edges, together with $0 \to 18$, cyclically connect all strongly connected components of the orbit digraph. Hence, the digraph $\Gamma_1(\mathcal{E}_{21})$ is strongly connected. By Proposition 1.1 the automaton $\mathcal{E}_{21}$ is perfectly reachable.

We have verified by brute force examination of all words in $\{a, b\}^*$ up to length 21 that none of them expand the subset $3 \oplus 7\mathbb{Z}_{21} = \{3, 10, 17\}$. The shortest word that expands $\{3, 10, 17\}$ is the word $ab^{14}ab^6$ of length 22.

Although the automaton $\mathcal{E}_{21}$ possesses a subset that is not 21-expandable, we have verified that it is not a counterexample to Don's conjecture. The only 'bad' subset $\{3, 10, 17\}$ turns out to be the image of the word

$$(ab^{15}ab^3ab^4)^2ab^4(ab^3(ab^4)^2)^2ab^3ab^4ab^7(ab^4)^2ab^{14}ab^6. \tag{3.2}$$

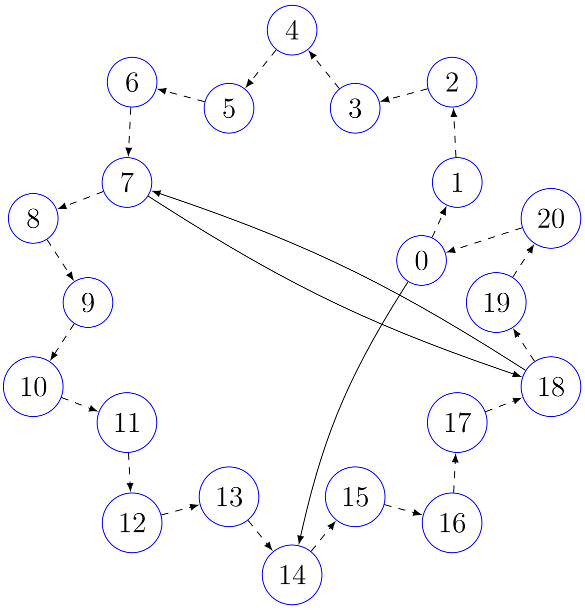


Figure 3.6: The automaton $\mathscr{E}_{21} = \langle \mathbb{Z}_{21}, \{a, b\} \rangle$. Solid and dashed edges show the action of $a$ and, resp., $b$; if $a$ fixes a state, the corresponding loop is omitted to improve readability.

The length of the word (3.2) is 132, which is much less than the bound $21(21-3) = 378$ claimed by Don's conjecture. The word (3.2) can be decomposed into a product of $18 = 21-3$ words of defect 1, but only the rightmost factor of defect 1 has length greater than 21 while all other factors are shorter, which by far compensates the excess of the last factor.

## 3.4 Additional remarks

Using the concept of expandability, we have confirmed Don's conjecture for standardized automata subject to an arithmetical restriction to their orbits. Moreover, we have proved that in every standardized automaton $\langle \mathbb{Z}_n, \{a, b\} \rangle$, almost all subsets are $n$-expandable; the only possible exceptions are the unions of cosets of the orbit subgroup of the automaton so that if the subgroup has index $k$ in $(\mathbb{Z}_n, \oplus)$, then at least $2^n - 2^k$ subsets of $\mathbb{Z}_n$ are $n$-expandable. On the other hand, we found an example of a 21-state perfectly reachable standardized automaton with a subset that is not 21-expandable.

To our surprise, our results reveal that the situation around the expandability approach to Don's conjecture for perfectly reachable automata is in parallel with that around the extensibility approach to Černý's conjecture for synchronizable automata. A method that has proved to be efficient for proving the Černý conjecture for special classes of automata is based on the following notion. For an automaton $\mathcal{A} = \langle Q, \Sigma \rangle$, a subset $S \subseteq Q$ and a word $w \in \Sigma^*$, denote by $Sw^{-1}$ the full preimage of $S$ under the action of $w$, that is, $Sw^{-1} := \{q \in Q \mid q \cdot w \in S\}$. A word $w \in \Sigma^*$ is said to *extend* a proper non-empty subset $S \subset Q$ if $|Sw^{-1}| > |S|$. Assuming that $|Q| = n$, we say that a proper non-empty subset $S \subset Q$ is *$n$-extensible* if $S$ can be extended by a word of length at most $n$ It is well known (and easy to see) that each $n$-state automaton, all of whose proper non-empty subsets are $n$-extensible, is synchronizable and has a reset word of length at most $(n-1)^2$; that is, it fulfills the Černý conjecture. The approach to the Černý conjecture via $n$-extensibility traces back to Jean-Éric Pin's paper [32]; the most striking applications of this approach are the proofs of the Černý conjecture for circular synchronizable automata (Lois Dubuc [16]) and synchronizable automata with Eulerian underlying graphs (Jarkko Kari [26]). On the other hand, Kari [25] found an example of a 6-state synchronizable automaton (see Figure 1) with a subset that is not 6-extensible. This shows that, in general, the Černý conjecture cannot be proved via $n$-extensibility.

Clearly, every completely reachable automaton is synchronizable. The converse is not true: it is easy to exhibit synchronizable, but not completely reachable automaton, even in the class of standardized automata. As a concrete instance, consider the standardized automaton shown in Figure 3.7; it has $ab^4ab(ab^2)^2aba$ as a reset word, while the subset $\{0, 1, 3, 4\}$ is not reachable.

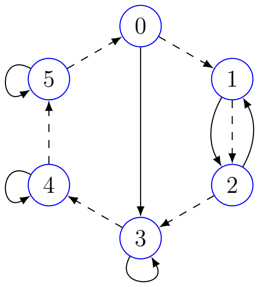


Figure 3.7: A standardized synchronizable automaton that is not completely reachable.

The relations between expanding and extending words are similar. It is easy to see that if, in an automaton $\mathcal{A} = \langle Q, \Sigma \rangle$, a word $w \in \Sigma^*$ expands a subset $S \subset Q$, then $w$ extends $S$ as well. The converse is not true, even for standardized automata. As an instance, we can reuse the automaton in Figure 3.7 where the word $ab^4ab(ab^2)^2aba$ extends the subset $\{0, 1, 3, 4\}$ but does not expand it. The automaton $\mathcal{E}_{21}$ shows that even a standardized perfectly reachable automaton $\langle \mathbb{Z}_n, \{a, b\} \rangle$ may have a subset that is not $n$-expandable, although all of its proper non-empty subsets are $n$-extensible by [16, Proposition 4.6].

Thus, the notions of a perfectly reachable automaton and expandability are specializations of those of a synchronizable automaton and extensibility, respectively. Nevertheless, these specialized concepts behave similarly to how their more general counterparts do. One can, therefore, speculate that further advances in studying perfectly reachable automata and Don's conjecture may contribute to a better understanding of synchronizable automata and progress towards resolving the Černý conjecture.

Let us come back to Zhu's work on [39]. There he achieves to prove that every standardized automaton $\langle \mathbb{Z}_n, \{a, b\} \rangle$ whose orbit subgroup coincides with the subgroup $2\mathbb{Z}_n$ of the group $(\mathbb{Z}_n, \oplus)$ fulfills Don's conjecture; this strengthens Theorem 3.1 of the present work. Additionally he reduces the bound of the words reaching non-empty subsets in standardized binary completely reachable automata to $n(n - k) + n - 1$ for all the subsets of size $k \geq 1$. In his paper Zhu takes the concepts and ideas presented in the last two chapters and apply ingenious arguments to obtain the aforementioned results.

# Chapter 4

# Completely Reachable Almost Group Automata

In this chapter we expand the work done in Chapter 2. We consider automata where all the letters except one are permutations of the set of states and the additional letter is of defect 1. The study of this kind of automata is by no means new; but they are presented with different names. In [2], they are called *almost-permutation automata*; and are used to present an example of a series of slowly synchronizable automata. In [3], automata are under the disguise of transformation semigroups and are called *near permutation.* In [6], they are called *almost-group automata*, there it is proved that these automata synchronize with high probability. Finally, in [34], the non-permutation letter acts as the identity for every state except for a subset where they have the same image. In the latter article, it is proved that if no equivalence relation is preserved, then the automaton is synchronizable. Among these papers, we would like to highlight the work done in [24] where the primitivity of a group of permutations of a state set has been tightly related to the complete reachability of the automata generated by adding a non permutation letter. Thus, the result presented here approaches this theory from the other side where the group is transitive but not primitive and looking for a condition when the automaton generated is completely reachable.

As we mentioned previously, in [24, Theorem 3.1] the following characterization of primitive permutation groups is given. Here $[n] := \{1, 2, \ldots, n\}$ and if $S$ is a set of transformations, then $\langle S\rangle$ is the transformation semigroup generated by $S$.

**Theorem 4.1** ([24]). *Let $G$ be a permutation group on $[n]$ with $n \geq 3$. Then $G$ is primitive if and only if for each transformation $f : [n] \to [n]$ of* defect 1 *every non-empty subset $A \subseteq [n]$ is reachable in $\langle G \cup \{f\}\rangle$.*

This theorem characterizes primitive groups. Moreover, it states that in presence of a primitive group, the addition of any transformation of defect 1 suffices to obtain a completely reachable automaton. Here we study the related case when the group produced is transitive but not primitive. We will see how this situation is not that forgiving and we need to ask more from the transformation of defect 1. The condition considered is very related to the one presented in Chapter 2 for automata with just one permutation letter and one with defect 1.

For the rest of this chapter we will consider automata $\mathcal{A} = \langle Q, \Sigma \rangle$, where $\Sigma = \Sigma_0 \cup \{a\}$ and:

- The set of letters $\Sigma_0 \subset S_Q$, are all permutations of $Q$.
- The generated subgroup $G = \langle \Sigma_0 \rangle$ is transitive.
- The letter $a$ has defect 1.

The excluded state of the only letter of defect 1 will be denoted by $e$, i.e., $\mathrm{excl}(a) = e$. Unless specified otherwise, the group generated by all permutation letters is denoted as $G$. We will denote automata with these characteristics as *almost group* automata.

Let $r \in \mathrm{coll}(a)$ be one of the two states collapsed by $a$. There is a permutation that sends $e$ to $r$ call it $\sigma \in G$. The transformation $\sigma a$ has defect 1, $e = \mathrm{excl}(\sigma a)$, and $e \in \mathrm{coll}(\sigma a)$. Consider the automaton $\overline{\mathcal{A}} = \langle Q, \Sigma_0 \cup \{\sigma a\} \rangle$. Note that $\mathcal{A}$ is completely reachable if and only if $\overline{\mathcal{A}}$ is completely reachable. Therefore, there is no loss of generality when we add the condition that $e \in \mathrm{coll}(a)$ from the beginning. When this happens we call the automaton *standardized*. This is similar to the process described in Chapter 2. This change will simplify the arguments we use for the rest of the chapter.

In this work when a block of imprimitivity is mentioned the trivial cases, i.e., the whole set or singletons, are not considered.

Recall that a subset of states, $P \subseteq Q$ is *invariant* by a transformation $w \in \Sigma^*$, or *$w$-invariant*, if $P \cdot w \subseteq P$. The condition to get complete reachability in the binary case is that no subset of states that *represents* a subgroup of the cyclic group is invariant by letter of defect 1. In the case of almost group automata this representation is not so clear. The cosets of any subgroup generate a partition of the group that contains this subgroup. There is a parallel situation in the case of blocks of imprimitivity, they form a partition of the set of states. There are subgroups for each of these blocks of imprimitivity that let them invariant[1]. This is the main reason to consider

[1] Considerations of this are treated ahead in the chapter

systems of imprimitivity. These partitions of the set of states are the closest to represent subgroups of the group acting on the states. Finally, for the binary case our standardization allowed the excluded state to be in every subgroup as the element labelled by 0. We state this detail more explicitly: the blocks of imprimitivity who are not $a$-invariant must be those that contain $\mathrm{excl}(a)$.

## 4.1 The necessary condition

Like in Chapter 2 we begin proving that complete reachability implies not $a$-invariance:

**Proposition 4.1.** *Let $\mathcal{A}$ be a standardized almost group automaton. If $\mathcal{A}$ is completely reachable, then $G$ is transitive and if there is at least a block of imprimitivity then no block of imprimitivity that contains $e = \mathrm{excl}(a)$ is invariant by $a$.*

*Proof.* First, let us prove that the condition for the group generated by the set of permutations to be transitive is necessary. For every word $w \in \Sigma^*$ it is true that $e \in \mathrm{excl}(wa)$; furthermore,

$$|\mathrm{excl}(w)| \leq |\mathrm{excl}(wa)| \leq |\mathrm{excl}(w)| + 1.$$

This is, the action of adding $a$ to a word either increases by one or keeps the defect of the resulting word. Note that adding a permutation does not modify the defect of any word. Hence, in order to reach the subsets $Q \setminus \{q\}$ for every $q \in Q$, it is necessary that there exists a permutation $\sigma_q \in G$ such that $e \cdot \sigma_q = q$. Let $p, q \in Q$ be an arbitrary pair of states. By the previously said, if $\mathcal{A}$ is completely reachable, then there are two permutations $\sigma_p, \sigma_q \in G$ such that $e \cdot \sigma_p = p$ and $e \cdot \sigma_q = q$. Finally, note that $p \cdot \sigma_p^{-1} \sigma_q = q$. Thus, $G$ is transitive.

For a subset of states $S \subset Q$, we denote by $\overline{S}$ its complement, i.e., $Q \setminus S$.

To prove that every block of imprimitivity that contains $e$ must be $a$-invariant, we proceed by contradiction; thus suppose that $B$ is a block of imprimitivity that contains $e$ and is $a$-invariant. This block belongs to a system of imprimitivity, let us call it $\mathfrak{B}$. Recall that $\mathcal{A}$ is completely reachable. Hence, for every block $C \in \mathfrak{B}$ there is a word that reaches its complement, i.e., there is $v \in \Sigma^*$ such that $Q \cdot v = \overline{C}$. Suppose that $w \in \Sigma^*$ is the shortest word that reaches one of theses complements. If $w = w' \, b$ with $b \in \Sigma_0$, then $Q \cdot w' = \overline{C} \cdot b^{-1}$, the complement of a block of imprimitivity in $\mathfrak{B}$. This contradicts the condition of $w$ being the shortest word. And we can conclude that $w$ does not end in a permutation.

Then, the word $w$ ends with the letter $a$, i.e., $w = w'a$. Recall that $Q \cdot w'a \subset Q \cdot a$ and $e \notin Q \cdot a$, thus $e \notin Q \cdot w$ and we conclude that $Q \cdot w = \overline{B}$.

Since $B$ is $a$-invariant, by our supposition, we can conclude that its complement is also $a$-invariant. And since every $q \in \overline{B}$ has a preimage by $a$ then this letter acts as a permutation of $\overline{B}$. Therefore $Q \cdot w' = Q \cdot w'a = \overline{B}$ what, again, contradicts the supposition of $w$ being the shortest. There is no other type of letter in which the word $w$ could finish, then we end with an absurd. This situation came from supposing that $B$ is $a$-invariant, thus we have our proposition. □

By the preceding proof we have:

**Corollary 4.1.** *If there is a block of imprimitivity that contains $e$ and is invariant by $a$, then its complement is not reachable.*

The assumption that $e \in \operatorname{coll}(a)$ is useful not only for the previous proof, but also for the correspondent arguments used in the rest of the chapter. Thus, we will maintain this condition henceforth.

## 4.2 Rystsov graphs of almost group automata

In this section we study the Rystsov graphs generated from almost group automata. We show that the blocks of imprimitivity keep playing a key role at the moment of studying these automata.

Let us use the following automaton as a running example to illustrate the construction of these graphs for the particular case we are studying in this chapter. Consider $\mathcal{E}_{18} := \langle \{1, 2, \ldots, 18\}, \{a, b, c\} \rangle$. The letters $b$ and $c$ are permutations with the following cyclic representation:

$$\begin{aligned} b :=&(1, 11, 13, 5, 7, 17)(2, 10, 14, 4, 8, 16)(3, 12, 15, 6, 9, 18) \\ c :=&(1, 3, 2)(4, 5, 6)(7, 13)(8, 16)(9, 15)(10, 14)(11, 17)(12, 18). \end{aligned}$$

Figure 4.1 represents the underlying graph of these letters, the action of $b$ and $c$ is shown with dashed and dotted edges respectively.

The transformation $a$ has defect 1. The following representation of $a$ puts the respective image under each state and omits the states that do not change:

$$\begin{pmatrix} 1\,2\,5\,6\,8 \\ 6\,8\,6\,5\,2 \end{pmatrix}.$$

Additionally, Figure 4.2 gives us a graphic representation of the action of the letter $a$ on the states where it does not act as the identity.

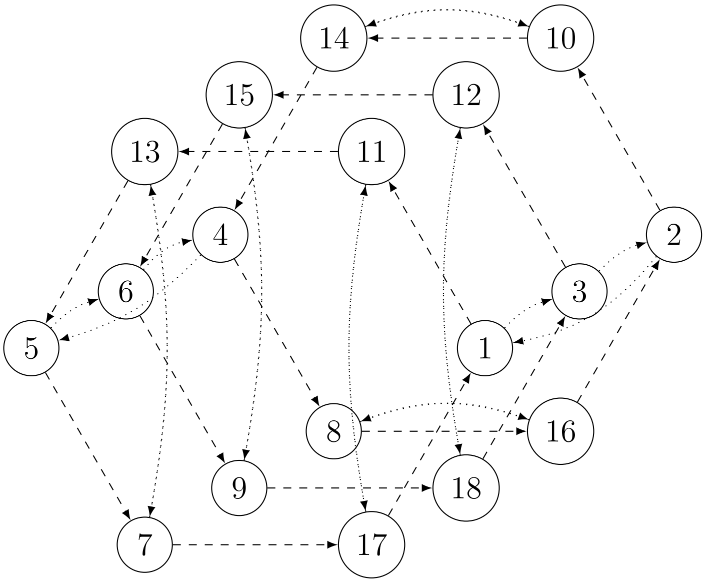


Figure 4.1: The action of the permutation letters $b$ and $c$ on the states of $\mathcal{E}_{18}$.

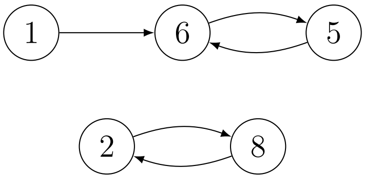


Figure 4.2: The action of the letter $a$ on some states of $\mathcal{E}_{18}$.

Note that the $\mathrm{excl}(a) = 1$, $\mathrm{dupl}(a) = 6$ and $\mathrm{coll}(a) = \{1, 5\}$. The group generated by $\{b, c\}$ is transitive and the blocks of imprimitivity that contain the state 1 are the sets

$$\{1, 5\}, \{1, 2, 3, 4, 5, 6\}.$$

Note that $1 \xrightarrow{a} 6$, $2 \xrightarrow{ac^2} 5$ and $1 \xrightarrow{ab^3a} 3$ are edges in $E_1(\mathcal{E}_{18})$.

For any automaton $\mathcal{A}$ recall that $W_1(\mathcal{A})$ is the set of all words in $\Sigma^*$ of defect 1. Also, consider the following subset of states:

$$D_1(\mathcal{A}) := \{p \in Q_1 \mid p = \mathrm{dupl}(w) \quad \& \quad e = \mathrm{excl}(w) \text{ for } w \in W_1(\mathcal{A})\}.$$

As in the case of binary automata these are the states directly connected to $e$ in $\Gamma_1(\mathcal{A})$, that is, targets of the edges with $e$ as source. The following lemma states that all the edges in $\Gamma_1(\mathcal{A})$ are images by $G$ of these initial edges.

**Lemma 4.1.** *If $q \to p \in E(\Gamma_1(\mathcal{A}))$, then there are $\sigma_q \in G$ and $d \in D_1(\mathcal{A})$ such that $e \cdot \sigma_q = q$ and $d \cdot \sigma_q = p$. Or, what is equivalent, there is a permutation $\sigma_q \in G$ such that $p \cdot \sigma_q^{-1} \in D_1(\mathcal{A})$.*

*Proof.* If $q \to p$ is an edge of $\Gamma_1(\mathcal{A})$, then there is a word $w \in W_1(\mathcal{A})$ of defect 1 such that $\mathrm{excl}(w) = q$ and $\mathrm{dupl}(w) = p$; this by definition of $\Gamma_1(\mathcal{A})$. Remember that $G$ is transitive, thus there is a permutation $\sigma_q \in G$ such that $e \cdot \sigma_q = q$.

The word $w\sigma_q^{-1}$ has defect 1 and $\mathrm{excl}(w\sigma_q^{-1}) = \mathrm{excl}(w) \cdot \sigma_q^{-1}$, at the same time $\mathrm{dupl}(w\sigma_q^{-1}) = \mathrm{dupl}(w) \cdot \sigma_q^{-1}$. Thus $\mathrm{excl}(w\sigma_q^{-1}) = q \cdot \sigma^{-1} = e$ and $p \cdot \sigma_q^{-1} \in D_1(\mathcal{A})$. □

This lemma also tell us that in order to compute $\Gamma_1(\mathcal{A})$ it is sufficient to calculate $D_1(\mathcal{A})$, and then apply to the generated edges permutations that send $e$ to each of the different states of the automaton. In our running example the initial edges of $\Gamma_1(\mathcal{E}_{18})$ are shown in Figure 4.3.

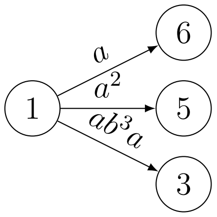


Figure 4.3: The initial edges of $\Gamma_1(\mathcal{E}_{18})$.

And the strongly connected component that contains 1 is shown in Figure 4.4 (we omitted the labels to lighten the picture).

Now, let $C_e^{[1]} \subseteq Q_1$ be the vertex set of the strongly connected component of $\Gamma_1(\mathcal{A})$ that contains $e$.

**Lemma 4.2.** *The set $C_e^{[1]}$ is a block of imprimitivity.*

*Proof.* If $\Gamma_1(\mathcal{A})$ is strongly connected then $C_e^{[1]} = Q$ and the proposition is true.

Let $\sigma \in G$ be the permutation such that $e \cdot \sigma = \mathrm{dupl}(a) = d$, then the edge $d \xrightarrow{a\sigma} d \cdot \sigma \in E_1$. If we repeat the application of $\sigma$, the resultant words all have defect 1, and there is an $i \geq 1$ such that $d \cdot \sigma^i = e$. Then, $C_e^{[1]}$ is not a singleton since at least $e, d \in C_e^{[1]}$.

Considering this we will prove first that for any $\sigma \in G$, $C_e^{[1]} \cdot \sigma$ is also a strongly connected component.

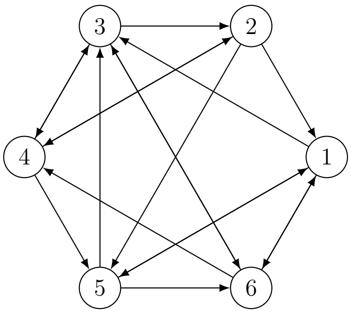


Figure 4.4: A strongly connected component of $\Gamma_1(\mathcal{E}_{18})$.

Let $p, q \in {C_e}^{[1]}$ be two arbitrary states. In $\Gamma_1(\mathcal{A})$ there is a path:

$$p \xrightarrow{w_1} t_1 \xrightarrow{w_2} t_2 \dots t_{k-1} \xrightarrow{w_k} q$$

where every $w_i$ is a word of defect 1. Since permutations act well on excluded and duplicated states, then:

$$p \cdot \sigma \xrightarrow{w_1 \sigma} t_1 \cdot \sigma \xrightarrow{w_2 \sigma} t_2 \cdot \sigma \dots t_{k-1} \cdot \sigma \xrightarrow{w_k \cdot \sigma} q \cdot \sigma$$

is a path in ${C_e}^{[1]} \cdot \sigma$; in the same way we can connect $q \cdot \sigma$ with $p \cdot \sigma$ making a strongly connected component.

If $x \in {C_e}^{[1]} \cap {C_e}^{[1]} \cdot \sigma$ let $y \in {C_e}^{[1]}$ and $z \in {C_e}^{[1]} \cdot \sigma$ be two different states. By the definition of a strongly connected component there are paths from $y$ to $x$, from $x$ to $z$, going back, from $z$ to $x$, and from $x$ to $y$. Then

$${C_e}^{[1]} = {C_e}^{[1]} \cdot \sigma.$$

This makes ${C_e}^{[1]}$ a block of imprimitivity. □

In the case that the group $G$ is primitive over $Q$, from Lemma 4.2 we can see that $\Gamma_1(\mathcal{A})$ will be strongly connected and by Theorem 1.1 it immediately follows that $\mathcal{A}$ is completely reachable. That is why, from now on we consider that the group $G$, besides being transitive, has at least a block of imprimitivity.

In the case of $\mathcal{E}_{18}$ we have seen that $C_e^{[1]} = \{1, 2, 3, 4, 5, 6\}$, and the other strongly connected components are the sets $B_2 = \{7, 8, 9, 10, 11, 12\}$ and $B_3 = \{13, 14, 15, 16, 17, 18\}$. Recall the description of Rystsov graphs detailed in Section 1.2. Since these sets have more than two elements, we can continue the construction of $\Gamma(\mathcal{E}_{18})$. Accordingly to the process of construction, the vertex set of $\Gamma_2(\mathcal{E}_{18})$ is $Q_2 = \{C_e^{[1]}, B_2, B_3\}$. Consider the word

$w := ab^3aca$, note that $\mathrm{excl}(w) = \{1, 3\}$ and $\mathrm{dupl}(w) = \{8, 6\}$, hence the edge $C_e^{[1]} \xrightarrow{w} B_2 \in E_2$. If we add $b$ twice more we have:

$$\mathrm{excl}(wb) = \{11, 12\}, \mathrm{dupl}(wb) = \{9, 16\}$$
$$\mathrm{excl}(wbb) = \{13, 15\}, \mathrm{dupl}(wbb) = \{18, 2\}.$$

Thus adding the edges $B_2 \xrightarrow{wb} B_3$ and $B_3 \xrightarrow{wbb} C_e^{[1]}$ to $E_2$. These are enough to conclude, thanks to Theorem 1.2, that $\mathcal{E}_{18}$ is completely reachable.

Now, we extend the results given by Lemma 4.1 and Lemma 4.2. Following the previous notation denote the strongly connected component that contains $e$ in the graph $\Gamma_k(\mathcal{A})$ as $C_e^{[k]}$. Recall the notions of *foliage* of a strongly connected component in a Rystsov graph introduced in Chapter 1. Most importantly, the foliage is the set of states inside the strongly connected component and is denoted by leaf().

**Lemma 4.3.** *If the foliages of the vertices in $\Gamma_k(\mathcal{A})$ form a system of imprimitivity of $G$ over $Q$, then $Y \to Z \in E_{k+1}$ if and only if there is a permutation $\sigma \in G$ and a set $X \in Q_k$ such that ${C_e}^{[k]} \to X \in E_{k+1}$, $\mathrm{leaf}(Y) = \mathrm{leaf}({C_e}^{[k]}) \cdot \sigma$ and $\mathrm{leaf}(X) \cdot \sigma = \mathrm{leaf}(Z)$.*

*Proof.* Since permutations respect the defect of any word and act well on excluded and duplicated sets, the converse is easy to see.

Now, if $Y \xrightarrow{w} Z \in E_{k+1}$, with $w \in W_{k+1}(\mathcal{A})$, then $\mathrm{excl}(w) \subset \mathrm{leaf}(Y)$ and $\mathrm{dupl}(w) \cap \mathrm{leaf}(Z) \neq \emptyset$. Let $w = u\,a\,\sigma$ with $\sigma \in G$, this is, $\sigma$ is the longest permutation after the last appearance of the letter $a$ in $w$. Since permutations do not increase the defect of a word, then $ua \in W_{k+1}(\mathcal{A})$ and $\mathrm{excl}(w\sigma^{-1}) = \mathrm{excl}(ua)$. From the last affirmation we can conclude that $\mathrm{excl}(ua) \subseteq \mathrm{leaf}(Y) \cdot \sigma^{-1}$.

By hypothesis, $\mathrm{leaf}(Y)$ is a block of imprimitivity then $\mathrm{leaf}(Y) \cdot \sigma^{-1}$ is one as well. Recall that $e \in \mathrm{excl}(ua)$ thus $\mathrm{excl}(ua) \subseteq {C_e}^{[k]} = \mathrm{leaf}(Y) \cdot \sigma^{-1}$. Using the same argument we can conclude that $\mathrm{leaf}(X) = \mathrm{leaf}(Z) \cdot \sigma^{-1}$. □

**Lemma 4.4.** *If the foliages of the vertices in $\Gamma_k(\mathcal{A})$ form a system of imprimitivity, then the foliage of ${C_e}^{[k+1]}$ is a block of imprimitivity of $G$ over $Q$.*

*Proof.* If each of the foliages of the vertices of $\Gamma_k(\mathcal{A})$ forms a system of imprimitivity, then the foliage of ${C_e}^{[k+1]}$ is just the union of blocks of imprimitivity. We can use an argument similar to the one used in the proof of Lemma 4.2 to prove that the image by any $\sigma \in G$ of the foliage of ${C_e}^{[k+1]}$ is also a strongly connected component and a block of imprimitivity. □

The previous lemmata form the proof by induction of:

**Proposition 4.2.** *For any $k \geq 1$, the foliages of the vertices of each $\Gamma_k(\mathcal{A})$ form a system of imprimitivity.*

Note that for any $k \geq 1$ if it happens that $\mathrm{leaf}({C_e}^{[k]}) = Q$ then $\Gamma_k(\mathcal{A})$ is strongly connected and $\mathcal{A}$ is completely reachable. Now we will proof that if this is not the case for any $k$, then some block of imprimitivity that contains $e$ is invariant by $a$. We will use the following set:

$$\begin{aligned} D_k(\mathcal{A}) := \{p \in Q \mid p \in \mathrm{dupl}(w) \text{ for some } w \in \Sigma^* \\ \text{such that } |\mathrm{excl}(w)| \leq k \text{ and } e \in \mathrm{excl}(w) \subseteq \mathrm{leaf}({C_e}^{[k-1]})\}. \end{aligned}$$

The set of states duplicated by words of defect less than $k$ such that their excluded set is contained in $C_e^{[k]}$.

## 4.3 Intermezzo

Before we continue, it is necessary to present some definitions and results related with the theory of permutation groups that are used in the rest of this chapter. Let $Q$ be a finite set and $G \leq S_Q$ be a group of permutations of $Q$. For any non-empty subset $P \subset Q$ consider the set of permutations:

$$\mathrm{St}_G(P) := \{\sigma \in G \mid P \cdot \sigma = P\}.$$

This is, the set of permutations of $G$ that preserve $P$ set-wise. It can be easily proved that $\mathrm{St}_G(P)$ is a subgroup of $G$, let us call it the *stabilizer* of the subset $P$.

Now consider an arbitrary but fixed system of imprimitivity of $G$ over $Q$, call it $\mathfrak{B}$. For the sake of completeness we present the proof of the following fact although it is well known.

**Proposition 4.3.** *Let $G$ be a group of permutations of the finite set $Q$. Suppose that $G$ is transitive and $\mathfrak{B}$ is a system of imprimitivity. If $B, C \in \mathfrak{B}$ are two different blocks of imprimitivity then* $\mathrm{St}_G(B)$ *and* $\mathrm{St}_G(C)$ *are conjugate subgroups of $G$.*

*Proof.* Let $p \in B$ and $q \in C$ arbitrary elements of these blocks of imprimitivity. The group $G$ is transitive over $Q$, that means there is a permutation $\sigma \in G$ such that $p \cdot \sigma = q$. By definition of blocks of imprimitivity it can be stated that $B \cdot \sigma = C$. Let us show that $\mathrm{St}_G(B) = \sigma^{-1}\mathrm{St}_G(C)\sigma$. First, if $\tau \in \mathrm{St}_G(C)$, then for any $r \in B$ it is true that $r \cdot \sigma^{-1}\tau\sigma \in B$; thus $\sigma^{-1}\mathrm{St}_G(C)\sigma \subseteq \mathrm{St}_G(B)$. For a similar argument we can say that $\sigma\mathrm{St}_G(B)\sigma^{-1} \subseteq \mathrm{St}_G(C)$, finishing our proof. □

The *core* of a subgroup $H$ of a group $G$, denoted by $\mathrm{Cr}(H)$, is the intersection of all the conjugates of $H$ in $G$, i.e.,

$$\mathrm{Cr}(H) := \bigcap_{\sigma \in G} \sigma^{-1} H \sigma.$$

Note that this subgroup is normal for $G$ and $H$ as well.

Resuming with the transitive group $G$ of permutations of $Q$, Proposition 4.3 tell us that for every system of imprimitivity $\mathfrak{B}$ of $Q$ all the stabilizers of the blocks are conjugate. Hence, it makes sense to talk about the core of $\mathfrak{B}$, or even the core of a block of imprimitivity $B \in \mathfrak{B}$ and denote them $\mathrm{Cr}(\mathfrak{B})$ and $\mathrm{Cr}(B)$ respectively. Moreover, we can give an alternative definition to the core of a system of imprimitivity as the intersection of all the stabilizers of its blocks. For our purposes we look for the core of certain blocks of imprimitivity to act in a transitive way on said blocks. We can ensure this if said core acts transitively on at least one of the blocks.

**Proposition 4.4.** *Let $G$ be a group of permutations of the finite set $Q$. Suppose that $G$ is transitive and $\mathfrak{B}$ is a system of imprimitivity. If $B \in \mathfrak{B}$ is a block and $\mathrm{Cr}(\mathfrak{B})$ acts transitively on $B$, then this core is also transitive on all the blocks of $\mathfrak{B}$.*

*Proof.* Let $C \in \mathfrak{B}$ be a different block of $\mathfrak{B}$, besides let $p, q \in C$ be two different states. We aim to prove that there is a permutation $\sigma \in \mathrm{Cr}(\mathfrak{B})$ such that $p \cdot \sigma = q$. Being $G$ transitive, there is a permutation $\tau \in G$ such that $C \cdot \tau = B$. Let $r, s \in B$ be such that $p \cdot \tau = r$ and $q \cdot \tau = s$. By hypothesis, there is a permutation $\rho \in \mathrm{Cr}(\mathfrak{B})$ such that $r \cdot \rho = s$. Thus

$$p \cdot \tau \rho \tau^{-1} = q.$$

Since the core is normal on $G$ we can conclude that $\tau \rho \tau^{-1} \in \mathrm{Cr}(\mathfrak{B})$. □

## 4.4 Non-reachability and invariance

In this part we see that for some almost-group automata not being completely reachable implies there is at least one block of imprimitivity invariant by the letter of defect 1.

Before the main proposition we present a technical lemma. Since in the following lemma $k$ is arbitrary but fixed, eventually $C_e^{[k]}$ will be referred just as $C_e$.

**Lemma 4.5.** *Let $\mathcal{A} = \langle Q, \Sigma_0 \cup \{a\} \rangle$ be an almost-group automaton. If in $\Gamma_\ell(\mathcal{A})$ there is an edge $C_e \to X$ and $\mathrm{Cr}(\mathrm{leaf}(C_e))$ is transitive for $C_e$,* *then for every state $q \in \mathrm{leaf}(X)$, there exists a word $v$ of defect $\ell$ such that* $\mathrm{excl}(v) \subset \mathrm{leaf}(C_e)$ *and* $q \in \mathrm{dupl}(v)$.

*Proof.* The edge $C_e \to X$ is produced by a word $w$ such that $\mathrm{excl}(w) \subset \mathrm{leaf}(C_e)$ and $\mathrm{dupl}(w) \cap \mathrm{leaf}(X) \neq \emptyset$. Let $p \in \mathrm{dupl}(w) \cap \mathrm{leaf}(X)$ be arbitrary. Since $\mathrm{Cr}(C_e)$ is transitive, by Proposition 4.4 there is a permutation $\sigma \in \mathrm{Cr}(C_e)$ such that $p \cdot \sigma = q$. At the same time it is true that $C_e \cdot \sigma = C_e$, since the core is a subset of $\mathrm{St}_G(C_e)$. Therefore we have that $\mathrm{excl}(w\sigma) \subset \mathrm{leaf}(C_e)$ and $q \in \mathrm{dupl}(w\sigma)$. As wanted. □

Using the Lemma 4.3 we also can conclude:

**Corollary 4.2.** *If in $\Gamma_k(\mathcal{A})$ there is an edge $X \xrightarrow{w} Y$ and* $\mathrm{Cr}(\mathrm{leaf}(C_e))$ *is transitive for $C_e$. Then for every state $q \in \mathrm{leaf}(Y)$, there exists a word $v$ of defect $k$ such that* $\mathrm{excl}(v) \subseteq \mathrm{leaf}(X)$ *and* $q \in \mathrm{dupl}(v)$.

With these two lemmas, we are ready for the main result of this part:

**Theorem 4.2.** *Let $\mathcal{A} = \langle Q, \Sigma_0 \cup \{a\} \rangle$ be an almost-group automaton. Suppose $\Gamma(\mathcal{A})$ is not strongly connected. This means for some $k \geq 1$ it happens that $\Gamma(\mathcal{A}) = \Gamma_k(\mathcal{A})$; and ${C_e}^{[k]} = {C_e}^{[j]}$ for every $j \geq k$. Besides this, suppose that for every $\ell \leq k$ the cores* $\mathrm{Cr}({C_e}^{[\ell]})$ *are transitive on ${C_e}^{[\ell]}$. Then* $\mathrm{leaf}({C_e}^{[k]})$ *is invariant by $a$.*

*Proof.* We will use a, structurally, similar proof of the fact for binary automata presented in the proof of Proposition 2.6. Suppose that ${C_e}^{[k]} = {C_e}^{[k+1]}$. By induction on $0 \leq \ell \leq k$ we will prove that

$$\mathrm{leaf}({C_e}^{[\ell]}) \cdot a \subseteq \mathrm{leaf}({C_e}^{[k]}).$$

For $\ell = 0$ take ${C_e}^{[0]} = \{e\}$ hence the proposition is true in this case.

Our *first induction hypothesis* is that $\mathrm{leaf}({C_e}^{[\ell]}) \cdot a \subseteq \mathrm{leaf}({C_e}^{[k]})$. By the construction of $\Gamma_{\ell+1}(\mathcal{A})$, for any $p \in \mathrm{leaf}({C_e}^{[\ell+1]})$ there is a $X_m \in Q_\ell$ such that $p \in \mathrm{leaf}(X_m)$ and there is a path:

$${C_e}^{[\ell]} \to X_1 \to X_2 \to \cdots \to X_m$$

in $\Gamma_\ell(\mathcal{A})$.

Now, by induction on the length of the path (the number $m > 1$) the idea is to prove that $\mathrm{leaf}(X_m) \cdot a \subseteq \mathrm{leaf}({C_e}^{[k]})$.

If $m = 1$, since there is an edge ${C_e}^{[\ell]} \to X_1$ we use Lemma 4.5 to ensure that for $p \in \mathrm{leaf}(X_1)$ there is a word $w \in W_\ell(\mathcal{A})$ such that $\mathrm{excl}(w) \subseteq$

$\mathrm{leaf}({C_e}^{[\ell]})$ and $p \in \mathrm{dupl}(w) \cap \mathrm{leaf}(X_1)$. The defect of $wa$ is at most $\ell + 1 \leq k + 1$ and by the *first induction hypothesis* $\mathrm{excl}(wa) \subseteq \mathrm{leaf}({C_e}^{[k]})$ and

$$p \cdot a \in \mathrm{dupl}(wa) \subseteq D_{k+1}(\mathcal{A}) \subseteq {C_e}^{[k+1]} = {C_e}^{[k]},$$

proving what we wanted.

Now suppose that $m > 1$ and $\mathrm{leaf}(X_{m-1}) \cdot a \subseteq \mathrm{leaf}({C_e}^{[k]})$, i.e., the *second induction hypothesis.* By the Corollary 4.2 for $p \in \mathrm{leaf}(X_m)$ there is a word $w \in W_\ell(\mathcal{A})$ such that $\mathrm{excl}(w) \subseteq \mathrm{leaf}(X_{m-1})$ and $p \in \mathrm{dupl}(w)$. If we apply the same argument as before, but this time using the *second induction hypothesis* we can conclude that

$$p \cdot a \in \mathrm{dupl}(wa) \subseteq D_{k+1}(\mathcal{A}) \subseteq {C_e}^{[k+1]} = {C_e}^{[k]},$$

again, as intended.

Since ${C_e}^{[\ell+1]}$ is a strongly connected component of $\Gamma_\ell(\mathcal{A})$, thus its foliage is the union of the respective foliages of its vertices. We have proved that $\mathrm{leaf}({C_e}^{[\ell+1]}) \cdot a \subseteq \mathrm{leaf}({C_e}^{[k+1]}) = \mathrm{leaf}({C_e}^{[k]})$. □

The condition of the core of the system of imprimitivty being transitive on the blocks of imprimitivity is important for the previous proof that emulates the argument presented in Proposition 2.6. This condition is present in all abelian groups, but not only in these, e.g., if $G$ is isomorphic to the dihedral group of $|Q|$ states. Note that the cyclic group, being abelian, satisfies this condition, thus our results generalize the case for binary completely reachable automata. The author conjectures that this condition can be dropped and it is possible to prove that if the Rystsov graph of an almost group automata is not strongly connected then there is a block of imprimitivity invariant by $a$. But more work must be done in this direction.

Since completely reachable binary automata are almost group automata, Zhu's counter examples (presented in Chapter 3) discard the possibility of the validity of Don's Conjecture for the latter. Nevertheless, the same remarks made in that moment can be done here: for the case the automaton is in standardized form the problem it is still open. The fact that the binary case has been proven to be a not trivial one indicates that for almost group automata more work awaits to be done.

# Chapter 5

# A Characterization of Totally Compatible Automata

Until now we have considered the reachability of non-empty subsets of states of an automaton. A completely reachable automaton has the capacity of “realizing” every non-empty subset of states with a word. Inspired by this it is natural to consider the dual concept: partitions of the state set. While the images of words define subsets, their kernels define partitions and it is possible to look for automata that realize every partition of the state set. This means that we focus not on the image set of the transformation (the *“right side”*), but on the partition defined by the transformation over the domain (the *“left side”*).

In this section we consider the class of automata such that every possible equivalence relation on the set of states can be obtained from a word. First we properly define them and give some examples. Then a characterization of these automata is given. Using this characterization we develop an algorithm to decide whether an automaton is totally compatible or not in polynomial time in the number of states. Following this algorithm we discuss some additional details of this type of automata.

## 5.1 Definition and examples

First, we define what exactly means that a word realizes an equivalence of the state set. Let $\mathcal{A} = \langle Q, \Sigma \rangle$ be an automaton and $w \in \Sigma^*$ a word of the automaton. Recall the definition of *kernel* of $w$ as the binary relation $\ker(w) \subset Q \times Q$ where $(p, q) \in \ker(w)$ if and only if $p \cdot w = q \cdot w$. This relation is not only an equivalence relation but it is preserved by all the letters of the

| Partition | Compatible Word |
|---|---|
| 1\|2\|3 | $c$ |
| 12\|3 | $t$ |
| 13\|2 | $ct$ |
| 1\|23 | $cct$ |
| 123 | $tct$ |

Table 5.1: Partitions of three elements and their compatible words.

automaton. Let $\rho$ be an arbitrary equivalence relation over the set $Q$. We say that a word $w \in \Sigma^*$ is *compatible* with $\rho$ if and only if $\ker(w) = \rho$. From this definition the following remark can be easily seen:

**Remark 5.1.** *Let $R_1, \ldots, R_k$, $k \geq 1$, be the classes of the equivalence $\rho$. A word $w \in \Sigma^*$ is compatible with $\rho$ if and only if:*

1. *The word synchronizes each class, i.e., $|R_i \cdot w| = 1$ for $1 \leq i \leq k$, and*
2. *the images by $w$ of different classes are pairwise different, i.e., $R_i \cdot w \neq R_j \cdot w$ for all $1 \leq i \neq j \leq k$.*

Note that any permutation (or the empty word) and a reset word are compatible with the trivial partitions, those made by just singletons and the whole set respectively.

**Definition 5.1.** The automaton $\mathcal{A} = \langle Q, \Sigma \rangle$ is **totally compatible** if for every equivalence relation $\rho$ over the set $Q$ there is a word $w_\rho \in \Sigma^*$ compatible with this relation.

An automaton whose transformations generate the full transformation semigroup is trivially totally compatible. Note that any totally compatible automaton is synchronizable.

The automaton $\mathcal{T} = \langle \{1,2,3\}, \{c,t\} \rangle$, where $c = (1,2,3)$, is the cyclic permutation, and the action of the letter $t$ is defined by: $1 \cdot t = 2 \cdot t = 1$ and $3 \cdot t = 3$ (see Figure 5.1), is an example of a totally compatible automaton. In Table 5.1 we can see all the partitions of three elements and their compatible transformations in $\mathcal{T}$.

Recall the sequence of Černý automata $\mathcal{C}_n$, with $n \geq 2$. We show in the following section that these are examples of completely reachable automata but not totally compatible.

From the previous information, we can make sense of Figure 5.2 which illustrates the subset relation between some relevant, for our discussion, kinds

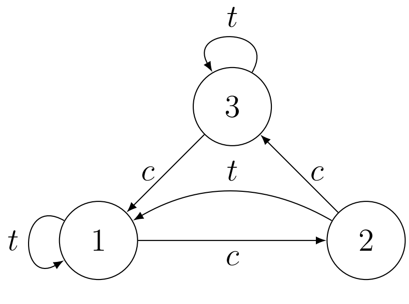


Figure 5.1: The automaton $\mathcal{T}$.

of automata. Superior classes of automata contain inferior ones. *Full transformation* automata are such that their transformation letters generate every possible transformation on the set of states.

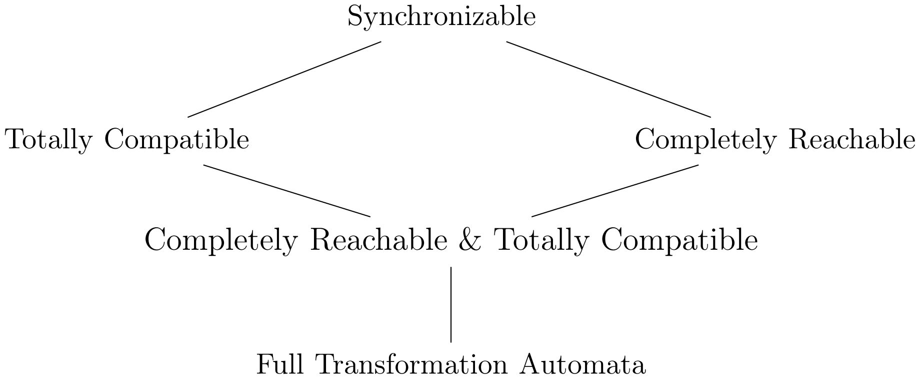

Synchronizable
Totally Compatible
Completely Reachable
Completely Reachable & Totally Compatible
Full Transformation Automata

Figure 5.2: Inclusion relation of some classes of automata.

Each of these continences are proper. In the next sections it is discussed in detail about completely reachable automata who are not totally compatible and vice versa. Let us briefly discuss about an automaton which is completely reachable and totally compatible but not full transformation. This example comes from [20, Section 3], in this article a series of automata with a pair of permutation letters is shown. Figure 5.3 represents the automaton $\mathcal{F}_7$ with 7 seven states.[1] We differentiate the permutation letters by dashed and solid edges. The main characteristic of this automaton that is interesting for us is the following: For every pair of subsets of states of size two $\{q_i, q_j\}$ and $\{q_k, q_m\}$ there are permutations $w$ and $v$ such that

$$\{q_i, q_j\} \cdot w = \{q_k, q_m\} \text{ and } \{q_k, q_m\} \cdot v = \{q_i, q_j\}.$$

[1] We deviate from our usual notation of naming the states with natural numbers in order to respect the notation used in the source.

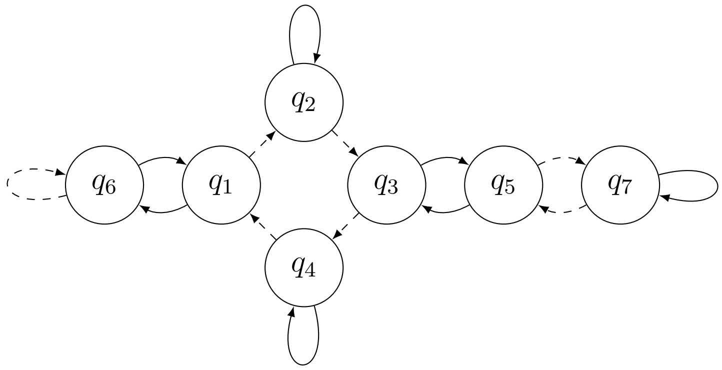


Figure 5.3: The automaton $\mathcal{F}_7$.

In other words, the graph of pairs is strongly connected. The next section clarifies why it is the case that if we add any letter of defect 1, the obtained automaton is totally compatible. In particular, if the letter of defect 1 collapses the subset $\{q_7, q_4\}$ to the state $q_7$ and fixes the other states, then a quick calculation shows that the Rystsov graph is strongly connected and the obtained automaton is completely reachable. Finally, note that both permutations are even, thus the group generated by then can not be the whole symmetric group of 7 elements and it is not possible to form all the possible transformations of 7 elements.

## 5.2 The characterization

**Theorem 5.1.** *The automaton $\mathcal{A} = \langle Q, \Sigma \rangle$ is totally compatible if and only if for every pair of states $p, q \in Q$ there is a word of defect 1 $w \in \Sigma^*$ such that $p \cdot w = q \cdot w$ or $\mathrm{coll}(w) = \{p, q\}$.*

*Proof.* The direct implication is rather easy to see. If $\mathcal{A}$ is totally compatible and $p, q \in Q$ are two arbitrary different states, then there is a word in $\Sigma^*$ compatible with the equivalence relation generated by adding the pairs $(p, q)$ and $(q, p)$ to the identity relation. This is a word of defect 1 that synchronizes these states.

On the other hand if $\mathcal{A}$ meets the condition, let $|Q| = n$. It will be shown that, for any $1 \leq k < n - 1$, if any equivalence relation of index (the number of different equivalence classes) larger than $k$ has a compatible word, then a word compatible with any equivalence relation of index $k$ can be constructed. This will prove the result since the empty word (or identity transformation) is compatible with the trivial equivalence of index $n$, and by hypothesis all the equivalence relations of index $n-1$ have compatible words. Suppose that any equivalence relation of index strictly bigger than $k \geq 1$ has a compatible word in $\Sigma^*$; and let $\rho$ be an arbitrary equivalence relation over $Q$ of index $k$ with $R_1, \ldots, R_k$ its classes. Without loss of generality suppose that $|R_k| > 1$. Now, let $\pi$ be an equivalence relation over $Q$ of index $k+1$ with equivalence classes $P_1, \ldots, P_k, P_{k+1}$ such that for $1 \leq i < k$ all its equivalence classes coincide with those of $\rho$, i.e., $P_i = R_i$; and in the case of $k$, $R_k = P_k \cup P_{k+1}$. This has sense since $R_k$ has more than one element. From our assumption there is a word $w_\pi \in \Sigma^*$ compatible with $\pi$. Note that both conditions of the Remark 5.1 are met by $w_\pi$ with almost all equivalence classes of $\rho$ except for $R_k$, where $R_k \cdot w_\pi = (P_k \cup P_{k+1}) \cdot w_\pi = \{p, q\}$ and $p \neq q$. By hypothesis, there is a defect 1 word $v \in \Sigma^*$ such that $p \cdot v = q \cdot v$, and acting as an injection in the rest of states. Observe that $w_\pi v$ synchronizes $R_k$ thus meeting both conditions of the Remark 5.1, and making it compatible with $\rho$. □

Note how the characterization is fairly similar to that of synchronizable automata: an automaton $\mathcal{A}$ is synchronizable if and only if every pair of states can be synchronized. The main difference is that in our case we ask the word that synchronizes has to be of defect 1. Also the proof of Theorem 5.1 suggests a method to construct a word compatible with any partition of the set of states. A word compatible with the trivial partition of all states in one class is a synchronizing word. From this fact it is fair to assume that the greedy algorithm that could be derived from the aforementioned proof would not produce an automaton's shortest reset word.

We can, already, show why Černý 's automata are not totally compatible. Let $n > 2$ be arbitrary but fixed and consider $\mathcal{C}_n$. The automaton $\mathcal{C}_n$ has $n$ states, $\{0, \ldots, n-1\}$ and two letters $\{a, b\}$, where $b$ acts as the complete cycle in the usual order and $a$ fixes every state except 0 that is sent to 1, i.e., $\mathrm{coll}(a) = \{0, 1\}$. Let $\{p, q\}$ and $\{r, s\}$ be different pairs of states. There is a $k \geq 1$ such that $\{p, q\} \cdot b^k = \{r, s\}$ if and only if:

$$p - q \equiv r - s \pmod{n} \quad \text{or} \quad p - q \equiv s - r \pmod{n}.$$

Because of this, we know there is no permutation such that

$$\{0, 2\} \cdot w = \{0, 1\},$$

thus there is no word of defect 1 with $\{0, 2\}$ as its collapsed set and $\mathcal{C}_n$ is not totally compatible.

## 5.3 Decidability, size and synchronization

From the stated in Theorem 5.1 we can derive some additional properties of totally compatible automata. The characterization suggests that in order to decide if an arbitrary automaton is totally compatible the focus must be on the letters that act as permutations and the ones which have defect 1. Since these transformations have the biggest ranks (cardinal of the image set), $n$ and $n-1$ respectively, and the composition of transformations does not increment the rank, it would not be possible to obtain transformations of defect 1 using transformations with lower ranks.

Using the previous terms an automaton is totally compatible if and only if every set of states of cardinality 2 (a 2-subset) is the collapsed set of some word of defect 1. Given two words of defect 1 of the same automaton, $w$ and $v$, recall that the word $wv$ keeps the defect if and only if $\mathrm{excl}(w) \in \mathrm{coll}(v)$, in this case $\mathrm{coll}(wv) = \mathrm{coll}(w)$; therefore the concatenation of two of words of defect 1 that keeps the defect will not create new collapsed sets.

To decide whether a given automaton is totally compatible or not we need the following construction. Let $\mathcal{A} = \langle Q, \Sigma \rangle$ be an automaton, denote by $\Sigma_0$ and $\Sigma_1$ the subsets of letters in $\Sigma$ with defect 0 and 1 respectively. Now consider the directed and labelled graph $\mathcal{A}_0^{[2]}$ with vertex set the 2-subsets of $Q$, denoted here by $Q^{[2]}$; and where $P \xrightarrow{b} T$, with $b \in \Sigma_0$, is a directed edge if and only if $P \cdot b = T$. Define:

$$\mathfrak{C} := \{P \in Q^{[2]} \mid P = \mathrm{coll}(a),\ \text{for some } a \in \Sigma_1\},$$

the subset of the collapsed sets of every letter in $\Sigma_1$. If there is a directed path in the graph $\mathcal{A}_0^{[2]}$, labelled by the sequence of letters $b_1, b_2, \ldots, b_k$ ($k \geq 0$), that begins at the 2-subset $P \in Q^{[2]}$ and finishes at some $\mathrm{coll}(a) \in \mathfrak{C}$, then the word $w = (b_1 b_2 \ldots b_k)\, a$ has defect 1. It is easy to see that $\mathrm{coll}(w) = P$.

From the previous discussion the procedure to decide whether $\mathcal{A}$ is shown in Algorithm 3.

**Algorithm 3** Decide whether an automaton is totally compatible

```
Input: An automaton $\mathcal{A} = \langle Q, \Sigma \rangle$.
Output: true if $\mathcal{A}$ is totally compatible, false otherwise.
Find $\Sigma_0$ and $\Sigma_1$.
Find $\mathfrak{C}$.
Construct the graph $\mathcal{A}_0^{[2]}$.
Find the strongly connected components of $\mathcal{A}_0^{[2]}$.
for all strongly connected components in $\mathcal{A}_0^{[2]}$ do
    Check whether there is at least one pair of $\mathfrak{C}$ in the strongly connected component.
if this is the case then
    return true
else
    return false
```

For an automaton with $n$ states $\mathcal{A}$, in order to generate $\mathcal{A}_0^{[2]}$ we need to consider the $\binom{n}{2}$ possible 2-subsets and for each one determine its image by each permutation in $\Sigma_0$. Tarjan's algorithm can be used to find the strongly connected components of $\mathcal{A}_0^{[2]}$. Both of these processes can be made in $O(n^2|\Sigma_0|)$ time. Finally, to check if every strongly connected component contains at least one pair of $\mathfrak{C}$ needs as many time as the number of strongly connected components, this is, $O(n^2)$. Thus, the decision problem of totally compatibility can be solved in polynomial time.

**Proposition 5.1.** *Totally compatible automata are **P**-decidable.*

Recall that if $w$ and $v$ are two words of defect 1 such that their concatenation, $wv$, has also defect 1 then $\text{coll}(wv) = \text{coll}(w)$, the collapsed set is preserved on the left. A consequence of the characterization and the aforementioned fact is that if there are not enough letters to collapse every pair of states there must be at least one letter that acts as a non-identity permutation. The fact that permutation groups that connect all possible 2-subsets of states (2-homogeneous) require at least two generators [4] makes us imply that if a totally compatible automaton has more than three states, then it must have more than three letters.

**Corollary 5.1.** *If $\mathcal{A} = \langle Q, \Sigma \rangle$ is a totally compatible automaton and $|Q| > 3$ then $|\Sigma| > 2$.*

For $k > 1$, a natural number, the *Bell's number*, $\mathcal{B}_k$ is the amount of possible partitions of a set with $k$ elements. From this, it is clear that the size of the syntactic monoid of any totally compatible automata with $n$ states is at least $\mathcal{B}_n$. It is natural to ask if this lower bound is reached by any totally compatible automaton, that is, for every possible partition of the state set, there is exactly one transformation in the syntactic monoid of the automaton with kernel equal to this partition. If this is the case we say that this automaton is *minimal.*

In [19, Section 12.4], it is reported the existence of a transformation semigroup over any finite set that contains exactly one transformation for every partition of that set. From this we can deduce the existence of a minimal totally compatible automaton. Moreover, up to isomorphism this semigroup is unique. Here we describe the transformations of this semigroup and a set of generators.

Let $Q$ be a finite set with $n$ elements and $\prec$ an arbitrary but fixed linear order of $Q$, i.e., $Q = \{q_1 \prec q_2 \prec \cdots \prec q_n\}$. Let $P, R \subset Q$ be non-empty and disjoint, we say that $P \prec R$ if and only if the minimum element of $P$, with respect to $\prec$, precedes the minimum element of $R$. Therefore any partition of $Q$ can be linearly ordered. For the arbitrary partition $P_1 \prec P_2 \prec \cdots \prec P_k$ of $Q$ define the transformation that sends the elements of the set $P_i$ to the state $q_i$ for $1 \leq i \leq k$. That is, the $i$-th subset in the ordered partition is sent to the $i$-th element in the order. Any automaton such that its transformation letters generate this semigroup is an example of a minimal totally compatible automaton.

Now, let us show a set of generators for this semigroup. For each possible 2-subset of states $\{q_i, q_j\} \subset Q$, where (without loss of generality) $q_i \prec q_j$, define the transformation letter $a_{i,j}$ by:

$$q_k \cdot a_{i,j} \colon = \begin{cases} q_k \text{ if } 1 \leq k < j; \\ q_i \text{ if } k = j; \\ q_{k-1} \text{ if } k > j. \end{cases}$$

It is easy to see that these transformations belong to the aforementioned semigroup and, following the same lines of the proof of Theorem 5.1, they generate it. Furthermore, note that the only permutation in this minimal semigroup is the identity, thus this is the smallest set of generators possible. It is worth noting that this minimal automaton is an example of a totally compatible automaton which is not completely reachable since the subsets $Q \setminus \{q_i\}$ for $1 \leq i < n$ are not reachable. For every $n > 1$ let us call the automaton with $n$ states and letters the generating set described before by $\mathcal{MTC}_n$.

Until now, we have seen examples of completely reachable but not totally compatible automata and vice versa. The following proposition gives a necessary condition for a totally compatible automaton to be completely reachable. The idea behind it was inspired by [3]. First recall that a transformation $u$, of defect 1 such that $\mathrm{coll}(u) = \{q, p\}$, is idempotent ($u^2 = u$) if and only if, without loss of generality, $p \cdot u = q$ and the transformation acts as the identity on the rest of states; hence $\mathrm{excl}(u) = p$ and $\mathrm{dupl}(u) = q$.

**Lemma 5.1.** *Let $\mathcal{A} = \langle Q, \Sigma \rangle$ be an automaton. Suppose that the group generated by $\Sigma_0$ (the transformation letters that act as permutations over $Q$) is transitive. Let $u \in \Sigma^*$ be a transformation of defect 1 with* $\mathrm{coll}(u) = \{q, p\}$*. Then $\Sigma^*$ contains both idempotents of defect 1 with collapsed set $\{q, p\}$.*

*Proof.* Let $e \in Q$ be the excluded state of the transformation $u$. Since the group generated by the permutations letters is transitive, then there is a permutation $\sigma \in \Sigma^*$ such that $e \cdot \sigma = q$. Consider the transformation $u\sigma \in \Sigma^*$. It has defect 1, $\mathrm{excl}(u\sigma) = q$ and $\mathrm{coll}(u\sigma) = \mathrm{coll}(u)$. Besides, $u\sigma$ acts as a permutation on the set $Q \setminus \{q\}$. Hence, there is a positive number $k > 1$ such that $(u\sigma)^k$ acts as the identity on this set. This transformation is still of defect 1 and $\mathrm{coll}((u\sigma)^k) = \mathrm{coll}(u)$ since collapsed sets do not change with the addition of suffixes. Furthermore $\mathrm{excl}((u\sigma)^k) = q$ and

$$p \cdot (u\sigma)^k = p = q \cdot (u\sigma)^k.$$

Thus $(u\sigma)^k$ is one of the idempotents we look for. A repetition of the previous argument now with $p$ as the image of the excluded set completes the proof. □

With this lemma we can now say the following about Rystsov graph of a totally compatible automaton with a transitive group.

**Proposition 5.2.** *Let $\mathcal{A} = \langle Q, \Sigma \rangle$ be a totally compatible automaton such that the group generated by $\Sigma_0$ is transitive. The graph $\Gamma_1(\mathcal{A})$ is complete on the vertex set $Q$.*

*Proof.* Let $p, q \in Q$ be two different states. Since $\mathcal{A}$ is a totally compatible automaton, by its characterization (Theorem 5.1) there is a transformation $w \in \Sigma^*$ of defect 1 such that $\mathrm{coll}(w) = \{p, q\}$. By the previous lemma, there are idempotent transformations of defect 1 such that $\{p, q\}$ is their collapsed set, thus these transformations define in $\Gamma_1(\mathcal{A})$ the edges $(p, q)$ and $(q, p)$. This happens for every arbitrary pair of states. Thus the graph is complete. □

By this we have the corollary:

**Corollary 5.2.** *If $\mathcal{A} = \langle Q, \Sigma \rangle$ is a totally compatible automaton such that the group generated by $\Sigma_0$ is transitive, then $\mathcal{A}$ is completely reachable.*

It is trivial to see that if an automaton is totally compatible, it is synchronizable since there must be a transformation for the complete equivalence relation. Thus, it is natural to ask for a bound of the shortest synchronization word of a totally compatible automaton. Using the naive method of synchronization by one pair at time it is easy to see that $\mathcal{MTC}_n$ synchronizes with a word of length $n$. Note that this automaton has a quadratic number of letters of defect 1 and no permutations. On the other hand if a totally compatible automaton has just one letter of defect 1, then it must have permutations letters that generate a 2-homogeneous, and therefore primitive, group [5]. Although not explicitly stated in [20, Remark 8] it is proven that automata where their permutation letters generate a 2-homogeneous group have a synchronization word of length at most quadratic on the number of states. These examples represent extreme cases in the number of letters of defect 1 of the automaton, between just one letter and a quadratic number of them. Work is yet to be done in the middle ground, where the group generated by the permutation letters does not connect all the 2-subsets of states and more than one letter of defect 1 is needed.

# Conclusion

In this work we have, mainly, dealt with complete reachable automata. This kind of automata are an special case of synchronizable ones. We narrowed our attention to automata whose letters have the largest possible ranks, this is, permutations and letters of defect 1. At first sight, this kind of automata looks simple, but they have shown a great potential for interesting research.

The main conclusions of this work are the results obtained in the development of it. Let us present a summary of these results.

- At first, extending the ideas given in [21], we proposed an algorithm to construct the Rystsov graph $\Gamma_k()$ of an arbitrary automaton with $n$ states and $m$ letters. This algorithm has a polynomial time complexity, with exponent $k$.

- We have characterized binary completely reachable automata. Our characterization leads to an algorithm that given a binary automaton $\mathcal{A}$, decides whether or not $\mathcal{A}$ is completely reachable in quasilinear time with respect to the size of $\mathcal{A}$. Our results heavily depend on the fact that apart from a single exception, binary completely reachable automata are *circular*, that is, have a letter acting as a cyclic permutation of the whole state set. Thus, the characterization depends on how the non permutation letter acts over the subsets of states that represent subgroups of the corresponding cyclic group.

- After characterizing binary completely reachable automata, we tackled the problem of bounding the length of the words that reach each subset. We found a partial characterization of standardized binary completely reachable automata that follow Don's conjecture. This was done by expanding Don's ideas and taking advantage of the characterization previously found. As noted in Chapter 3 the requirement to be a standardized automaton is not an innocuous one.

- Once considered the binary case we wanted to continue the discussion on complete reachability for automata with more than two letters. For this, we considered almost group automata. For them we had a partial success expanding the results obtained for the binary case. These results tell us that what is important for complete reachability is how the letter of defect 1 acts on the systems of blocks of the group generated by the permutations of the automaton. We managed to prove the necessity of the condition and a weaker version for the sufficient direction.

- Inspired by the notion of automata that can obtain every possible nonempty subset of states as image of some word, we continued our discussion with the dual notion: automata that have every possible partition of the states set as the kernel of some word. It was, also, proposed an algorithm that runs in polynomial time in the number of states to decide whether a finite automaton is of that type or not.

## Open problems and future work

In the process of researching and solving questions, most of the cases, new problems to be solved arise. Our work is not an exception of this. This is a list of some of the questions and problems we found. It is by no means complete and more inquiries may be waiting to be discovered.

- This work contains some examples of binary completely reachable automata whose Rystsov graph are disconnected when considering words of defect 1 and 2. It is yet to be found examples for bigger defects. By this we mean binary completely reachable automata whose Rystsov graphs $\Gamma_1, \Gamma_2$ and $\Gamma_3$ are disconnected but once we consider $\Gamma_4$ the graph becomes strongly connected. Off course the problem can be extended for an arbitrary large $k > 1$. To be more concrete: *Find binary completely reachable automata such that all the Rystsov graphs $\Gamma_i$, with $i \leq k$ are disconnected but the graph $\Gamma_{k+1}$ is connected.*

- Related to the previous problem it comes the problem of minimality. Recall that Corollary 2.1 allows us to conclude that the minimal number of states for binary completely reachable automata automata with $\Gamma_1$ disconnected is 12. We conjecture that in the case of $\Gamma_2$ being disconnected, the minimal number of states required is 48. Again, this problem can be extended to the general case.

- Zhu in [39] proved that Don's conjecture is not true for arbitrary completely reachable automata and gave bound for standardized binary completely reachable automata. Nevertheless it is still open if standardized binary completely reachable automata fulfill Don's conjecture or there are counter examples as in the general case. Although the expansion method alone is not enough more work in that direction could lead to some progress.

- The transitivity of the cores over the blocks is a condition necessary for the proof of Theorem 4.2. The author thinks this condition can be omitted; yet it is still open to find a proof for this fact. Adding to this, the aforementioned discussion about Don's conjecture and Rystsov graphs applies in this case for almost group automata.

- There is plenty of work to be done in regard of the length of synchronization words of totally compatible automata. Table 5.2 depicts the lower and upper bounds of synchronizing words for the different kinds of automata studied in this work. Note how for totally compatible automata both bounds are open and we present just the bounds given by some kind of automata that are contained and contain the kind of totally compatible.

| Kind of Automata | Lower bound | Upper bound |
|---|---|---|
| Synchronizable | $(n-1)^2$ | $0.1654n^3 + O(1)$[35] |
| Completely Reachable | $(n-1)^2$ | $2n^2 - n\ln(n) - 4n + 2$ [17] |
| Totally Compatible | $\frac{n(n-1)}{2}$[20] | $0.1654n^3 + O(1)$[35] |
| Completely Reachable and Totally Compatible | $\frac{n(n-1)}{2}$[20] | $2n^2 - n\ln(n) - 4n + 2$ [17] |
| Full Transformation Automata | $\frac{n(n-1)}{2}$[20] | $2n^2 - 6n + 5$[20] |

Table 5.2: Table with the bounds for the length of synchronizing words of some kind of automata.

The research on completely reachable automata was, partly, initiated to further study the synchronization problem and the Černý's conjecture. In the same vein totally compatible automata were conceived. But, the subject has gained inertia by itself and it has developed into a source of new and interesting problems and results. We hope this work is an evidence of this.

# Bibliography

[1] D. Ananichev and M. Volkov. Some results on Černý type problems for transformation semigroups. In *Proceedings of the Workshop Semigroups and Languages: Lisboa, Portugal, 27–29 November 2002*, pages 23–42. World Scientific, 2004.

[2] D. Ananichev and V. Vorel. A new lower bound for reset threshold of binary synchronizing automata with sink. *Journal of Automata, Languages and Combinatorics*, 24(2-=4):153–164, 2019.

[3] J. André. Near permutation semigroups. In *Proceedings of the Workshop Semigroups and Languages: Lisboa, Portugal, 27–29 November 2002*, pages 43–53. World Scientific, 2004.

[4] J. Araújo, W. Bentz, and P. Cameron. Orbits of primitive $k$-homogenous groups on $(n-k)$-partitions with applications to semigroups. *Transactions of the American Mathematical Society*, 371(1):105–136, 2018.

[5] R. Beaumont and R. Peterson. Set-transitive permutation groups. *Canadian Journal of Mathematics*, 7:35–42, 1955.

[6] M. Berlinkov and C. Nicaud. Synchronizing almost-group automata. *International Journal of Foundations of Computer Science*, 31(08):1091–1112, 2020.

[7] J. Berstel, D. Perrin, and C. Reutenauer. *Codes and Automata.* Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2009.

[8] E. Bondar, D. Casas, and M. Volkov. Completely reachable automata: An interplay between automata, graphs, and trees. *International Journal of Foundations of Computer Science*, 34(06):655–690, 2023.

[9] E. Bondar and M. Volkov. Completely reachable automata. In *Descriptional Complexity of Formal Systems*, pages 1–17. Springer, 2016.

[10] E. Bondar and M. Volkov. A characterization of completely reachable automata. In *Developments in Language Theory*, pages 145–155. Springer, 2018.

[11] D. Casas and M. Volkov. Binary completely reachable automata. In *LATIN 2022: Theoretical Informatics*, pages 345–358. Springer, 2022.

[12] D. Casas and M. Volkov. Don's conjecture for binary completely reachable automata: an approach and its limitations, 2024. Preprint on: https://arxiv.org/abs/2311.00077.

[13] J. Černỳ. Poznámka k homogénnym experimentom s konečnỳmi automatmi. *Matematicko-fyzikálny Časopis Slovensky Akadmie Vied*, 14(3):208–216, 1964.

[14] J. Černý. A note on homogeneous experiments with finite automata. *Journal of Automata, Languages and Combinatorics*, 24(2–4):123–132, 2019.

[15] H. Don. The Černý conjecture and 1-contracting automata. *The Electronic Journal of Combinatorics*, 23(3): article no. P3.12, 2016.

[16] L. Dubuc. Sur les automates circulaires et la conjecture de Černý. *RAIRO – Theoretical Informatics and Applications*, 32(1-3):21–34, 1998.

[17] R. Ferens and M. Szykuła. Completely reachable automata: A polynomial algorithm and quadratic upper bounds. In *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, pages 59:1–59:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.

[18] M. Fredman and D. Willard. Surpassing the information theoretic bound with fusion trees. *Journal of Computer and System Sciences*, 47(3):424–436, 1993.

[19] O. Ganyushkin and V. Mazorchuk. *Classical Finite Transformation Semigroups*. Springer, 2009.

[20] F. Gonze, V. Gusev, R. Jungers, B. Gerencsér, and M. Volkov. On the interplay between Černý and Babai's conjectures. *International Journal of Foundations of Computer Science*, 30(01):93–114, 2019.

[21] F. Gonze and R. Jungers. Hardly reachable subsets and completely reachable automata with 1-deficient words. *Journal of Automata, Languages and Combinatorics*, 24(2–4):321–342, 2019.

[22] G. Hardy and E. Wright. *An Introduction to the Theory of Numbers.* Oxford University Press, 6th edition, 2008.

[23] S. Hoffmann. Binary and circular automata having maximal state complexity for the set of synchronizing words. *Information and Computation*, 295: article no. 105076, 2023.

[24] S. Hoffmann. New characterizations of primitive permutation groups with applications to synchronizing automata. *Information and Computation*, 295: article no. 105086, 2023.

[25] J. Kari. A counter example to a conjecture concerning synchronizing words in finite automata. *Bulletin of the EATCS*, 73:146, 2001.

[26] J. Kari. Synchronizing finite automata on Eulerian digraphs. *Theorerical Computer Science*, 295:223–232, 2003.

[27] C. Liu. *Some memory aspects of finite automata.* PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering, 1962.

[28] M. Maslennikova. Reset complexity of ideal languages. In Mária Bieliková, Gerhard Friedrich, Georg Gottlob, Stefan Katzenbeisser, and György Turán, editors, *SOFSEM 2012: Theory and Practice of Computer Science (Institute of Computer Science Academy of Sciences of the Czech Republic)*, pages 33–44. See also https://arxiv.org/abs/1404.2816 , 2012.

[29] E. Moore. Gedanken-experiments on sequential machines. In *Automata Studies*, pages 129–154. Princeton University Press, 1956.

[30] B. Natarajan. An algorithmic approach to the automated design of parts orienters. In *27th Annual Symposium on Foundations of Computer Science*, pages 132–142, 1986.

[31] B. Natarajan. Some paradigms for the automated design of parts feeders. *The International Journal of Robotics Research*, 8(6):98–109, 1989.

[32] J. Pin. Sur un cas particulier de la conjecture de Cerny. In Giorgio Ausiello and Corrado Böhm, editors, *Automata, Languages and Programming, Fifth Colloquium, Udine, Italy, July 17–21, 1978, Proceedings*, pages 345–352, Springer, 1978.

[33] I. Rystsov. Estimation of the length of reset words for automata with simple idempotents. *Cybernetics and Systems Analysis*, 36(3):339–344, 2000.

[34] I. Rystsov and M. Szykuła. Primitive automata that are synchronizing, 2023. Preprint on https://arxiv.org/abs/2307.01302.

[35] Y. Shitov. An improvement to a recent upper bound for synchronizing words of finite automata. *Journal of Automata, Languages and Combinatorics*, 24(2–4):367–373, 2019.

[36] P. Starke. Eine Bemerkung über homogene Experimente. *Elektronische Informationsverarbeitung und Kybernetik*, 2(4):257–259, 1966.

[37] P. Starke. A remark about homogeneous experiments. *Journal of Automata, Languages and Combinatorics*, 24(2–4):133–137, 2019.

[38] M. Volkov. Synchronization of finite automata. *Russian Mathematical Surveys*, 77(5):819–891, 2022.

[39] Yinfeng Zhu. Around Don's conjecture for binary completely reachable automata. In Joel D. Day and Florin Manea, editors, *Developments in Language Theory*, pages 282–295, Springer, 2024.