Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics
Departement of Computer Science

# Equiloaded Automata

Master's Thesis

Ivan Kováč

**Supervisor:**
prof. RNDr. Branislav Rovan, PhD.

Bratislava, 2010

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

# Equiloaded Automata

Master's Thesis

**Study Program**: Informatics
**Branch of Study**: 9.2.1 Informatics
**Supervisor**: prof. RNDr. Branislav Rovan, PhD.

Bratislava, 2010

Bc. Ivan Kováč

I hereby declare that I wrote this thesis by myself, only with the help of the referenced literature, under the careful supervision of my thesis advisor.

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

# Acknowledgments

I am deeply grateful to my supervisor professor Branislav Rovan for his invaluable help and for providing a compelling subject for this thesis.

I want to thank my second cousin Erik Goldman for the corrections he provided, Mišo Kotrbčík for his advice, and Ondrej Budáč for help with the golden ratio.

Finally, I would like to sincerely thank my family and my girlfriend Ivana Selečéniová for their support and help.

# Abstract


| | |
|---|---|
| Author: | Ivan Kováč |
| Title: | Equiloaded Automata |
| University: | Comenius University in Bratislava |
| Faculty: | Faculty of Mathematics, Physics and Informatics |
| Departement: | Departement of Computer Science |
| Supervisor: | prof. RNDr. Branislav Rovan, PhD. |
| Number of Pages: | 37 |
| Year: | 2010 |



In this thesis we initiate the study of a balanced use of resources in computations. We consider a particular model of computation — deterministic finite automata — and take states as the resource to be used in a balanced way. In this setting we develop notions and prove results which can serve as an example for similar studies in other settings. Three possible approaches to define a balanced use of states by deterministic finite automaton are investigated: *a strict equiloadedness*, *an equiloadedness*, and *an equiloadedness on sequences of words*. We analyze properties of families of automata and languages with respect to different definitions of balanced use of states.

We show a characterization of the family of languages for which there exists a strictly equiloaded automaton. We exhibit the closure properties of this family based on this characterization.

The family of languages for which there exists an equiloaded automaton is analyzed by proving closure properties, by providing a necessary condition for a language to be in this family, and by defining a set of transformations that preserve the equiloadedness of an automaton.

Considering equiloadedness on sequences of words, we analyze the influence of different orderings of words on the equiloadedness tolerance. We investigate the equiloadedness on sequences for various bounds on the equiloadedness tolerance function.



KEYWORDS: equiloaded automata, balanced use of resources, deterministic finite automata

# Contents

**Abstract** **iv**

**Contents** **v**

**List of Figures** **vi**

**List of Tables** **vi**

**1 Introduction** **1**

**2 Preliminaries** **3**

**3 Strictly Equiloaded Automata** **5**
3.1 Definition and Equivalent Form . . . . . . . . . . 5
3.2 Elementary Results . . . . . . . . . . 7
3.3 Characterization . . . . . . . . . . 7
3.4 Closure Properties . . . . . . . . . . 8

**4 Equiloaded Automata** **10**
4.1 Definition and Equivalent Form . . . . . . . . . . 10
4.2 Elementary Results . . . . . . . . . . 12
4.3 Relationship with Minimal-State Automaton . . . . . . . . . . 14
4.4 Automata Transformations . . . . . . . . . . 17
4.5 Closure Properties . . . . . . . . . . 20
4.6 Characterization . . . . . . . . . . 24
4.7 Algebraic Representation . . . . . . . . . . 24
4.8 Open Questions . . . . . . . . . . 27

**5 Automata Equiloaded on Sequences of Words** **30**
5.1 Definition . . . . . . . . . . 30
5.2 Equivalent Form . . . . . . . . . . 31
5.3 Impact of Order of Words on Equiloadedness Tolerance . . . . . . . . . . 32

**6 Conclusion** **35**

**Bibliography** **36**

**Abstrakt** **37**

# List of Figures


3.1 A strictly equiloaded and a non-strictly equiloaded automaton . . . . . 6
3.2 An automaton for the language $L = \{\mathrm{aaba}, \mathrm{acba}\}^* \cdot \{\mathrm{ac}, \mathrm{aa}\}$ . . . . . . 8

4.1 An equiloaded and a non-equiloaded automaton . . . . . . . . . . . . . . 11
4.2 The minimal-state automaton for the language $L = \{u\mathrm{bb}v \mid u, v \in \Sigma^*\}$ 13
4.3 The minimal-state and an equiloaded automaton for $L = \{ab, ba\}^*$ . . . 15
4.4 An $A, B$-composition automaton . . . . . . . . . . . . . . . . . . . . . . . . 18
4.5 The minimal-state automaton for the language $\{\mathrm{a}, \mathrm{b}\}^* - \{\mathrm{aa}\}$. . . . . . 23
4.6 Two automata for the language $L = \{\mathrm{aaa}, \mathrm{bbb}\}^*$ . . . . . . . . . . . . . . 29


# List of Tables


4.1 Closure properties of $\mathcal{R}$, $\mathcal{L}_{\mathrm{EQA}}$ and $\mathcal{L}_{\mathrm{SEQA}}$ . . . . . . . . . . . . . . . . . 23

# Chapter 1

# Introduction

Our research is motivated by balancing resources in computations. A balanced use of resources can be important in many real-world problems. For example, it is desirable that all parts of a system (e.g., processor chip) are used so that none of the parts wears out substantially faster (or heats up more) than other parts.

Since this topic has not been investigated yet, it is necessary to develop the basic notions for this type of study. We have chosen a simple model of computation, deterministic finite automaton, and its natural resource, the state.

Deterministic finite-state automaton is a well known computation model. It has good properties for our research, for example on a word $w$, there are always $|w| + 1$ usages of states. Many problems in automata theory are at first studied on this simple model and then extended to more complex models such as pushdown automata, linear-bounded automata or even Turing machines.

We define a new property of finite automata, *equiloadedness*: the balanced use of each state of an automaton. We shall explore three ways of defining this notion.

### The Structure of Thesis

In Chapter 2 we define a deterministic finite automaton and related concepts, which is a computational model used throughout the thesis.

In Chapter 3 we start with a basic definition by which an automaton is strictly equiloaded if it uses every state equally often (except for a constant difference) on every word from the language accepted. We are able to characterize all strictly equiloaded automata, see Section 3.3.

In Chapter 4 we introduce a different definition of an equiloaded automaton — an automaton is equiloaded if, for each given length $n$, it uses its states equally often during computations in total over words of length $n$ from the language accepted. We describe a set of automata transformations, which do not change the equiloadedness property. We provide a necessary condition for a language to be in the family of languages accepted by equiloaded automata. We prove closure properties of this family. Finally, we conjecture a sufficient condition for an automaton to be equiloaded. If it holds, this conjecture would lead to infinite number of non-trivial equiloaded automata.

In Chapter 5 we analyze equiloadedness on sequences of words. This setting is motivated by a batch processing of an input. We do not require that an automaton uses its states equally often on all words. We rather determine some checkpoints where the states must be used equally often, up to the equiloadedness tolerance. In this chapter we answer questions about the impact of different orderings of words on equiloadedness on sequences of words.

We conclude by stating several open problems about equiloadedness and indicate several possibilities for continuing this research.

# Chapter 2

# Preliminaries

In this chapter, we present the definition of deterministic finite automaton (DFA) used in this thesis. We use the definition of DFA such that an automaton can halt without reading an input word to the end.

We use the halting DFA because we are only interested in the load of states while processing words from the accepted language. Words not from the accepted language, and thus rejected by the DFA, are not relevant to our analysis. The "dead state" used to finish reading input words will thus not distort the balanced use of other states. It is possible to study the balanced use of resources on non-halting DFA, but we find this setting more interesting.*

Throughout the thesis, we assume that the set of natural numbers $\mathbb{N}$ contains 0.

**Definition 2.1.** *A deterministic finite automaton* $A$ is a 5-tuple $A = (Q, \Sigma, \delta, q_0, F)$ consisting of a finite set of states $Q$ with an initial state $q_0 \in Q$, a finite alphabet $\Sigma$, a transition function $\delta : Q \times \Sigma \to Q$, and a set of accepting states $F \subseteq Q$.

*Remark.* In our version of DFA (halting DFA), we shall consider the transition function to be a partial function.

**Definition 2.2.** *A configuration* of the DFA $A$ is a pair $(q, w) \in Q \times \Sigma^*$, where $q$ is a state of the automaton and $w$ is the remaining part of the input.

**Definition 2.3.** *A computation step* of the DFA $A$ is a relation $\vdash_A$ on configurations defined by

$$(q, av) \vdash_A (p, v) \overset{\text{def}}{\iff} p = \delta(q, a).$$

*Remark.* If it is clear from the context which automaton we mean, we shall use $\vdash$ instead of $\vdash_A$. We shall denote the reflexive and transitive closure of $\vdash_A$ by $\vdash_A^*$.

**Definition 2.4.** *A language* accepted by the deterministic finite automaton $A$ is the set of words $L(A) = \{w \in \Sigma^* \mid (q_0, w) \vdash_A^* (q, \varepsilon),\ q \in F\}$.

*For strict equiloadedness used in Chapter 3 it is not hard to see that languages for which there is a strict equiloaded automata form "trivial" family of languages – all over one letter alphabet and of very specific form.

Some of our results are based on a graphical representation of a DFA. For the sake of completeness we shall define the term graphical representation.

**Definition 2.5.** Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA. *A graphical representation* of the DFA $A$ is the directed labelled graph $G(V, E)$, $V = Q$, with a marked set of vertices $F$ and an *initial* vertex $q_0$, where a set of arcs is defined as follows:

- There is an arc $(p, q) \in E$ labeled by $x$ if and only if $\delta(p, x) = q$ for some $x \in \Sigma$. Moreover, the number of arcs from $p$ to $q$ is equal to the number of different $x \in \Sigma$ such that $\delta(p, x) = q$.

*Remark.* We shall draw the non-accepting states in the graphical representation of $A$ as circles and accepting states as double circles. The initial vertex $q_0$ shall be marked with one additional input arc. The reader can see a graphical representation of a DFA at Figure 3.2 in the next chapter.

We shall use the term minimal-state automaton in respect to Myhill-Nerode Theorem. [Nerode, 1958]

**Theorem 2.1** (Myhill-Nerode)**.** Let $L \subseteq \Sigma^*$ be a language. The following statements are equivalent.

1. $L$ is a regular language.

2. There is a right-invariant equivalence relation $\sim$ of finite index such that $L$ is a union of some of the equivalence classes of $\sim$.

3. A relation $\sim_L$ defined by $u \sim_L v \iff (\forall x \quad ux \in L \iff vx \in L)$ is of finite index.

**Definition 2.6.** Let $L$ be a regular language. An automaton $A = (Q, \Sigma_L, \delta, q_0, F)$ is said to be *the minimal-state automaton* for the language $L$, if it is defined as follows. The states of $A$ are the equivalence classes of $\sim_L$ with $\varepsilon \in q_0$. The set of accepting states consists of equivalence classes such that $L = \bigcup_{q \in F} q$. The transition function $\delta$ is defined by

$$\delta(p, x) = q \overset{\text{def}}{\iff} \forall w \in p \quad wx \in q.$$

# Chapter 3

# Strictly Equiloaded Automata

In this chapter, we shall analyze properties of strictly equiloaded automata as defined in [Kováč, 2008]. We define the concept of strictly equiloaded automata. We show two results from [Kováč, 2008], relationship with minimal-state automaton and strict equiloadedness of finite languages. Finally, we characterize the family of languages for which there exists a strictly equiloaded automaton. Using this characterization we prove closure properties of the family of languages accepted by equiloaded automata.

## 3.1 Definition and Equivalent Form

A very strict view of a balanced use of states is that on every word in the accepted language all states are used equally many times. We shall relax this condition by an equiloadedness tolerance or equiloadedness constant* in such a way, that difference between load of any two states is less than the equiloadedness tolerance.

*Notation.* Let $A$ be a DFA and $w$ be a word from $L(A)$. We denote by $\#_A[q, w]$ (or simply by $\#[q, w]$) the number of times the automaton $A$ uses the state $q$ when processing the input $w$.

**Definition 3.1** (Strict equiloadedness)**.** Let $A = (Q, \Sigma, \delta, q_0, F)$ be a deterministic finite automaton. The automaton $A$ is *strictly equiloaded* at words from $L = L(A)$ if there exists $k \in \mathbb{R}^+$ such that

$$\forall w \in L,\ \forall p, q \in Q \quad |\#[p, w] - \#[q, w]| \leq k.$$

The smallest $k$ for which this inequality holds will be called an *equiloadedness constant.*

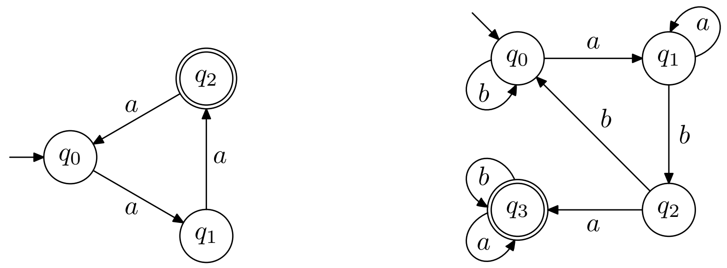
*Remark.* Instead of writing "$A$ is strictly equiloaded at words from $L(A)$" we shall simply write "$A$ is strictly equiloaded".

*Notation.* The family of languages for which there is some strictly equiloaded automaton is denoted by $\mathcal{L}_{\mathrm{SEQA}}$.

*In Chapter 5 we shall use the term equloadedness tolerance for a function, thus for the sake of clarity we shall now use the term equiloadedness constant

*Example* 3.1. The language $L = \{a^{3i+2} \mid i \in \mathbb{N}\}$ belongs to the family $\mathcal{L}_{\text{SEQA}}$. In Figure 3.1(a) we can see the minimal-state automaton $A$ for the language $L$. Because all computations on words from $L(A)$ end in the state $q_2$, every state is equally loaded on any $w \in L(A)$. Hence, the equiloadedness constant for $A$ is $k_A = 0$.

On the other hand, there is no strictly equiloaded automaton for the language $L' = \{w\text{aba}v \mid w, v \in \Sigma^*\}$. In Figure 3.1(b) is the minimal-state automaton $B$ that accepts this language. It is easy to prove non-equiloadedness of the automaton $B$ — on words $\text{aba}v \in L'$ is the accepting state $q_3$ used $|v|$ times and the state $q_0$ only once, thus there is no constant $k$ such that, for any length $n = |v|+3$, it holds $||v|-1| \leq k$. It can be proved from Theorem 3.1 that no strictly equiloaded minimal-state automaton implies non-existence of strictly equiloaded automaton.




(a) The automaton $A$ (b) The automaton $B$

Figure 3.1: An example of (a) a strictly equiloaded and (b) a non-strictly equiloaded automaton

Now, we shall provide an equivalent form of the condition of strict equiloadedness from the above definition.

**Lemma 3.1.** A DFA $A = (Q, \Sigma, \delta, q_0, F)$ is strictly equiloaded at words from $L = L(A)$ if and only if there exists $k_1 \in \mathbb{R}^+$ such that

$$\forall w \in L \quad \left| \max_{q \in Q}(\#[q, w]) - \frac{|w| + 1}{|Q|} \right| \leq k_1. \tag{3.1}$$

*Remark.* The fraction $(|w| + 1)/|Q|$ expresses the average load of states on the word $w$, since there are $|w| + 1$ states used and $|Q|$ is the number of states. Hence, when deciding whether an automaton $A$ is equiloaded, it suffices to compute the average load and compare it with the load of the most used state.

*Proof.* If $A$ is strictly equiloaded, for all $w \in L(A)$ it holds $|\#[p, w] - \#[q, w]| \leq k$. Hence, if we choose $p$ to be the most used state and $q$ to be the least used state, it holds that

$$\forall w \in L(A) \quad \left| \max_{q \in Q}(\#[q, w]) - \frac{|w| + 1}{|Q|} \right| \leq \left| \max_{q \in Q}(\#[q, w]) - \min_{q \in Q}(\#[q, w]) \right| \leq k$$

Conversely, if (3.1) holds, then the load of the most used state $q_{max}$ is bounded from above by $(|w| + 1)/|Q| + k_1$ and the load of the least used state $q_{min}$ is bounded from below by $(|w| + 1)/|Q| - k_1 \cdot (|Q| - 1)$. Thus, for a difference between the load of any two states it holds that $|\#[p, w]| - \#[q, w]| \leq |\#[q_{max}, w] - \#[q_{min}]| \leq k_1|Q| = k$. □

## 3.2 Elementary Results

For the sake of completeness we shall present two theorems from [Kováč, 2008]. The first theorem provides a strong principle for proving the strict equiloadedness of an automaton.

**Theorem 3.1.** A regular language $L$ is in $\mathcal{L}_{\text{SEQA}}$ if and only if the minimal-state automaton for $L$ is strictly equiloaded.

*Proof.* A proof can be found in [Kováč, 2008]. □

**Theorem 3.2.** Every finite language $L$ is in the family $\mathcal{L}_{\text{SEQA}}$.

*Proof.* Consider the minimal-state automaton $A$ for the language $L$. The automaton $A$ has no cycles, hence on every word is each state used once or not at all. Thus $|\#[p, w] - \#[q, w]| \leq 1$. □

## 3.3 Characterization

**Theorem 3.3** (Characterization of $\mathcal{L}_{\text{SEQA}}$)**.** Every language $L \in \mathcal{L}_{\text{SEQA}}$ is either finite or a graphical representation of any equiloaded automaton that accepts $L$ is an oriented multicycle through all states.

*Remark.* By an oriented multicycle through all states we mean an automaton with transition function defined by $\delta(q_i, x) = q_{(i+1) \bmod k}$, where $x \in \Sigma$. For the state $q_i$, it is possible that there exists more than one symbol from $\Sigma$ for which transition function is defined. See example 3.2.

*Proof.* Suppose that $L$ is an infinite language accepted by an equiloaded automaton $A = (Q, \Sigma, \delta, q_0, F)$. We will show that $A$ must contain a multicycle through all states. If it does not contain one, we know that $A$ is either acyclic (thus, $L(A)$ is a finite language) or there is a cycle $C$ which does not contain all states. Let $u$ be a substring such that, when $A$ processes $u$, $A$ changes its state from $q_0$ to $q_c$, where $q_c$ lies in the cycle $C$. Let $v$ be a substring such that $A$ (when it processes $v$) starts at the state $q_c$, visits all states from the cycle, and returns to the state $q_c$. Let $w$ be a substring such that $A$ changes its state from $f_c$ to an accepting state. If such a substring does not exist, then the load of $q_c$ on every word is equal to zero. Therefore $A$ is not strictly equiloaded.

If $w$ exists, then at word $uv^kw$ the DFA $A$ will not be strictly equiloaded. This implies that in $A$ must exist a multicycle through all states.

Now suppose that there exists a transition at symbol $x$, such that $A$ will change its state from (current) $q_i$ to $q_j$, where $j \neq i+1$. It means that there is another cycle in $A$, but not through all of its states. So we can use the same principle of constructing input for $A$ as above. It can be shown that $A$ is not a strictly equiloaded automaton, a contradiction. Therefore, a graphical representation of $A$ is a multicycle through all states. □

*Example* 3.2. The automaton $A = (\{q_0, q_1, q_2, q_3\}, \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}, \delta, q_0, \{q_2\})$ shown in Figure 3.2, where $\delta$ is defined by

$$\delta(q_0, \mathrm{a}) = q_1, \quad \delta(q_1, \mathrm{a}) = \delta(q_1, \mathrm{c}) = q_2,$$
$$\delta(q_2, \mathrm{b}) = q_3, \quad \delta(q_3, \mathrm{a}) = q_0,$$

is a multicycle. Thus, the language $L = \{\mathrm{aaba}, \mathrm{acba}\}^* \cdot \{\mathrm{ac}, \mathrm{aa}\}$ is strictly equiloaded.

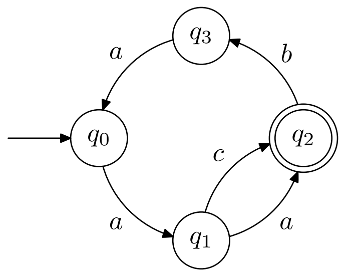



Figure 3.2: The automaton $A$ for $L = \{\mathrm{aaba}, \mathrm{acba}\}^* \cdot \{\mathrm{ac}, \mathrm{aa}\}$

*Example* 3.3. Based on our characterization of the family $\mathcal{L}_{\mathrm{SEQA}}$ it is easy to see that there are many regular languages not in this family. For example, any language of words that contain some substring, e.g., $L = \{w\mathrm{abba}v \mid w, v \in \Sigma^*\}$. It is unfortunate, but we can not expect that all regular language will be (strictly) equiloaded.

## 3.4 Closure Properties

As we shown in [Kováč, 2008], languages accepted by strictly equiloaded automata are not closed under inverse homomorphism, union, Kleene star, Kleene plus, concatenation and complement. Now we shall examine three additional closure properties. In particular, we show that $\mathcal{L}_{\mathrm{SEQA}}$ is not closed under homomorphism and reversal, and it is closed under intersection. Two of this results follows directly from the characterization of languages accepted by strictly equiloaded automata provided by Theorem 3.3.

**Theorem 3.4.** The family $\mathcal{L}_{\mathrm{SEQA}}$ is not closed under homomorphism.

*Proof.* Let $A = (\{q_0\}, \{\mathrm{a}, \mathrm{b}\}, \delta, q_0, \{q_0\})$ be a DFA, where $\delta(q_0, \mathrm{a}) = \delta(q_0, \mathrm{b}) = q_0$. The language accepted by the automaton $A$ is $L = L(A) = \{\mathrm{a}, \mathrm{b}\}^*$. Consider a homomorphism $h$ defined by $h(\mathrm{a}) = \mathrm{aab}$ and $h(\mathrm{b}) = \mathrm{aa}$. We would show that the language

$h(L(A)) = \{\mathrm{aa}x \mid x \in \{\varepsilon, \mathrm{b}\}\}^*$ is not equiloaded. The minimal-state automaton for this language is $B = (\{q_0, q_1, q_2\}, \{\mathrm{a}, \mathrm{b}\}, \delta_B, q_0, \{q_2\})$, where the transition function is defined by $\delta_B(q_0, \mathrm{a}) = q_1$, $\delta_B(q_2, \mathrm{a}) = q_1$, $\delta_B(q_1, \mathrm{a}) = q_2$, and $\delta_B(q_2, \mathrm{b}) = q_0$. The automaton $B$ is not equiloaded. (For example, at the word $h(\mathrm{b}^k) = \mathrm{a}^{2k}$.) □

**Theorem 3.5.** The family $\mathcal{L}_{\mathrm{SEQA}}$ is not closed under reversal.

*Proof.* Consider the language $L = \{(\mathrm{abb})^i x \mid x \in \{\mathrm{a}, \varepsilon\},\ i \in \mathbb{N}\}$. The graphical representation of the minimal-state automaton $A_1 = (\{q_0, q_1, q_2\}, \Sigma, \delta_1, q_0, \{q_0, q_1\})$ for $L$ is an oriented multicycle, hence $L \in \mathcal{L}_{\mathrm{SEQA}}$. (The transition function $\delta_1$ is defined as follows. $\delta_1(q_0, \mathrm{a}) = q_1$, $\delta_1(q_1, \mathrm{b}) = q_2$ and $\delta_1(q_2, \mathrm{b}) = q_0$.) Yet, the minimal-state automaton $A_2 = (Q, \Sigma, \delta_2, q_0, \{q_0, q_1\})$ for $L^R$ is not strictly equiloaded. The automaton $A_2$ has four states, and the transition function $\delta_2$ is defined by

$$\begin{aligned} &\delta_2(q_0, \mathrm{a}) = q_1, \quad \delta_2(q_0, \mathrm{b}) = q_2, \quad \delta_2(q_1, \mathrm{b}) = q_2, \\ &\delta_2(q_2, \mathrm{b}) = q_3, \quad \delta_2(q_3, \mathrm{a}) = q_1. \end{aligned}$$

Therefore, the graphical representation of $A_2$ is not an oriented multicycle and $A_2$ is not strictly equiloaded. □

**Theorem 3.6.** The family $\mathcal{L}_{\mathrm{SEQA}}$ is closed under intersection.

*Proof.* If at least one of $L_1, L_2 \in \mathcal{L}_{\mathrm{SEQA}}$ is finite, the intersection $L_1 \cap L_2$ is finite and thus is in $\mathcal{L}_{\mathrm{SEQA}}$. Let $A$, $B$ be automata accepting $L_1$ and $L_2$ respectively. If we use the standard construction† of an automaton from the intersection of two languages, then the result (call it $C$) will be either a multicycle (suppose we remove states that $C$ never visits) or $L(C)$ will be finite. If $L(C)$ is finite, then $L(C)$ is also strictly equiloaded. Suppose that $L(C)$ is infinite. If there is a state $(p, q)$ such that there exist states $(p_1, q_1)$ and $(p_2, q_2)$ such that

$$\delta_C((p, q), x) = (p_1, q_1), \quad \delta_C((p, q), y) = (p_2, q_2).$$

Thus $p_1 = p_2$ and $q_1 = q_2$ (both $A$ and $B$ are multicycles). Similarly, if there is a state $(p, q)$ such that there exist states $(p_1, q_1)$ and $(p_2, q_2)$ such that

$$\delta_C((p_1, q_1), x) = (p, q), \quad \delta_C((p_2, q_2), y) = (p, q),$$

then $p_1 = p_2$ and $q_1 = q_2$. Thus, $C$ is a multicycle and equiloaded. □

By above theorems and results in [Kováč, 2008], we can summarize the closure properties of $\mathcal{L}_{\mathrm{SEQA}}$. The family $\mathcal{L}_{\mathrm{SEQA}}$ is closed under intersection and it is not closed under homomorphism, inverse homomorphism, union, concatenation, Kleene star, Kleene plus, reversal and complement. The overview of closure properties of families $\mathcal{R}$, $\mathcal{L}_{\mathrm{SEQA}}$ and $\mathcal{L}_{\mathrm{EQA}}$‡ is shown in Table 4.1.

† States of the new automaton are pairs of states from $A$ and $B$ respectively. There is a transition from $(p, q)$ to $(p', q')$ on a symbol $x$ if and only if $\delta_A(p, x) = p'$ and $\delta_B(q, x) = q'$.

‡ Defined in the next chapter.

# Chapter 4

# Equiloaded Automata

In this chapter, we shall discuss a relaxation of the strict definition of an automaton which uses its states in a balanced way in total over all words of each given length $n$. The relaxation of the property of strict equiloadedness gives us new non-trivial automata which are equiloaded. We shall define some transformations of automata which preserve the equiloadedness property and analyze properties of equiloaded automata and the family of languages accepted by equiloaded automata.

## 4.1 Definition and Equivalent Form

*Notation.* Let $A$ be a DFA and $L'$ be a language, $L' \subseteq L(A)$. We denote by $\#_A[q, L']$ (or simply by $\#[q, L']$, if it is clear from the context which automaton we mean) the number of times the automaton $A$ uses the state $q$ when processing the inputs from $L'$. It holds that

$$\#_A[q, L'] = \sum_{w \in L'} \#_A[q, w].$$

**Definition 4.1** (Equiloadedness). Let $A = (Q, \Sigma, \delta, q_0, F)$ be a deterministic finite automaton. The automaton $A$ is *equiloaded* at words from $L = L(A)$ if there exists $k \in \mathbb{R}^+$ such that

$$\forall n \in \mathbb{N},\ \forall p, q \in Q \quad |\#[p, L \cap \Sigma^n] - \#[q, L \cap \Sigma^n]| \leq k|L \cap \Sigma^n|.$$

The smallest $k$ for which this inequality holds will be called the *equiloadedness constant.*

*Remark.* The factor $|L \cap \Sigma^n|$ on the right side of the inequality expresses the possibility of one state to be used constantly more than other on every word. Without this factor, there will be a strictly equiloaded automaton which is not equiloaded.

*Notation.* The family of languages $L$ for which there is an equiloaded automaton will be denoted by $\mathcal{L}_{\text{EQA}}$.

*Example* 4.1. Consider the language $L = \{\mathrm{a}^i\mathrm{b}^j \mid i, j \in \mathbb{N}\}$. The minimal-state automaton $A$ for this language shown in Figure 4.1(a). This automaton is equiloaded, as is shown in the proof of Theorem 4.1. Intuitively, for each word $\mathrm{a}^i\mathrm{b}^j$ there is a word $\mathrm{a}^j\mathrm{b}^i$ of the same length and automaton $A$ uses each state approximately $i + j$ times.

In Figure 4.1(b) there is a non-equiloaded automaton $B$. It can be shown that the load of the state $q_0$ on words of a given length $n = 3\ell$ is $(\ell + 1)|L \cup \Sigma^n|$. The average load of states on words of the length $3\ell$ is $(3\ell + 1)|L \cup \Sigma^n|/5$. From Lemma 4.1 and from above we can prove that $B$ is not equiloaded, because for any constant $k$ there is $\ell$ such that $(\ell + 1) - (3\ell + 1)/5 > k$. However, this does not mean that the language $L(B) = \{\mathrm{aaa}, \mathrm{bbb}\}^*$ is not equiloaded.


(a) An equiloaded automaton $A$ (b) A non-equiloaded automaton $B$

Figure 4.1: Example of (a) an equiloaded and (b) a non-equiloaded automaton

Similarly to the strict equiloadedness case, we provide an equivalent form of our definition. This form is mostly used in proofs of equiloadedness of an automaton because it is easier to consider the most used state rather than compare all possible pairs of states.

**Lemma 4.1.** A DFA $A = (Q, \Sigma, \delta, q_0, F)$ is equiloaded at words from $L = L(A)$ if and only if there exists $k_1 \in \mathbb{R}^+$ such that

$$\forall n \in \mathbb{N} \quad \left| \max_{q \in Q} (\#[q, L \cap \Sigma^n]) - \frac{(n+1) \cdot |L \cap \Sigma^n|}{|Q|} \right| \leq k_1 |L \cap \Sigma^n|. \tag{4.1}$$

*Remark.* The left side of the inequality (4.1) can be read as a difference between the load of the most used state and the average load of states at all words from $L$ of a given length $n$. For all $n$ where $L \cap \Sigma^n \neq \emptyset$, the inequality (4.1) can be written as

$$\frac{\max_{q \in Q} (\#[q, L \cap \Sigma^n])}{|L \cap \Sigma^n|} - \frac{n+1}{|Q|} \leq k_1.$$

Seeing that in the case where $L \cap \Sigma^n \neq \emptyset$, the inequality holds, we shall use the above forms of the equiloadedness condition as needed.

*Proof.* If $A$ is equiloaded, it holds that

$$\forall n \in \mathbb{N} \quad \left| \max_{q \in Q} (\#[q, L \cap \Sigma^n]) - \min_{q \in Q} (\#[q, L \cap \Sigma^n]) \right| \leq k |L \cap \Sigma^n|.$$

Since the least used state is used at most as many times as the average load of states, the inequality (4.1) is valid.

Conversely, suppose that the inequality (4.1) holds. For a given $n$, let $q_{max}$ be the most used state. A load of each state at words from $L \cap \Sigma^n$ is bounded from above by $\#[q_{max}, L \cap \Sigma^n]$ and therefore by

$$\frac{(n+1)|L \cap \Sigma^n|}{|Q|} + k_1|L \cap \Sigma^n|.$$

On the other hand, the load of every state is bounded from below by

$$\frac{(n+1)|L \cap \Sigma^n|}{|Q|} - k_1|L \cap \Sigma||(Q| - 1).$$

If we use these bounds in $|\#[p, L \cap \Sigma^n] - \#[q, L \cap \Sigma^n]|$, we obtain

$$|\#[p, L \cap \Sigma^n] - \#[q, L \cap \Sigma^n]| \leq k_1|L \cap \Sigma^n| + k_1|L \cap \Sigma^n|(|Q| - 1) \leq k_1|Q| \cdot |L \cap \Sigma^n|.$$

Since $k = k_1|Q|$ is a constant, $A$ is equiloaded. □

## 4.2 Elementary Results

**Theorem 4.1.** $\mathcal{L}_{\text{SEQA}} \subsetneq \mathcal{L}_{\text{EQA}} \subsetneq \mathcal{R}$.

*Proof.* Consider $L \in \mathcal{L}_{\text{SEQA}}$. There exists a strictly equiloaded automaton $A$ such that $L = L(A)$. Since $|\#[p, w] - \#[q, w]| \leq k$ for all $p, q \in Q$, for all $n$ it holds that

$$\begin{aligned} |\#[p, L \cap \Sigma^n] - \#[q, L \cap \Sigma^n]| &= \left| \sum_{w \in L \cap \Sigma^n} (\#[p, w] - \#[q, w]) \right| \\ &\leq \sum_{w \in L \cap \Sigma^n} |\#[p, w] - \#[q, w]| \leq k|L \cap \Sigma^n|. \end{aligned}$$

We showed that $\mathcal{L}_{\text{SEQA}} \subseteq \mathcal{L}_{\text{EQA}}$. To show that $\mathcal{L}_{\text{SEQA}} \neq \mathcal{L}_{\text{EQA}}$ it suffices to show that a language $L = \{\text{a}^i\text{b}^j \mid i, j \in \mathbb{N}\}$ is in $\mathcal{L}_{\text{EQA}}$. Let $A = (\{q_0, q_1\}, \{\text{a}, \text{b}\}, \delta, q_0, \{q_0, q_1\})$ be the minimal-state automaton for the language $L$. It is easy to see that the minimal-state automaton $A$ for $L$ is not strictly equiloaded, for its graphical representation is not an oriented multicycle. The transition function $\delta$ is defined by

$$\delta(q_0, \text{a}) = q_0, \quad \delta(q_0, \text{b}) = q_1, \quad \delta(q_1, \text{b}) = q_1.$$

We will show that the expression $|\#[q_0, L \cap \Sigma^n] - \#[q_1, L \cap \Sigma^n]|$ is bounded from above by $n+1 = |L \cap \Sigma^n|$. The load of the state $q_0$ is $\sum_{i=0}^{n} i+1$, because $\#[q_0, \text{a}^i\text{b}^{n-i}] = i+1$.

The load of the state $q_1$ is $\sum_{i=0}^{n} n-i = \sum_{i=0}^{n} i$. Therefore,

$$|\#[q_0, L \cap \Sigma^n]| - \#[q_1, L \cap \Sigma^n]| = \sum_{i=0}^{n}(i+1-i) = \sum_{i=0}^{n} 1 = n+1.$$

Hence, $A$ is equiloaded and $L \in \mathcal{L}_{\text{EQA}}$.

To show that $\mathcal{L}_{\text{EQA}} \subsetneq \mathcal{R}$ it suffices to find a language $L$ and its minimal-state automaton $B$ such that, for any $\ell$, there exists arbitrarily large $n$ such that $B$ uses a particular state more than $\ell$ times the load of another state on words of the length $n$. This statement is proved in Theorem 4.3.* Such a language is for example the language $L = \{w\text{bb}v \mid w, v \in \Sigma^*\}$ with the minimal-state automaton $B$ shown in Figure 4.2.

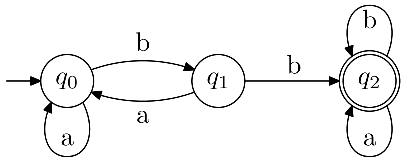



Figure 4.2: The minimal-state automaton $B$ for the language $L = \{u\text{bb}v \mid u, v \in \Sigma^*\}$

For a given length $n$, words from $L(A)$ are of the form $\text{a}^{i_1}\text{ba}^{i_2}\text{b}\ldots\text{a}^{i_m}\text{bb}v$, where $|v| = n - m - 1 - \sum_{j=0}^{m} i_j$. The load of $q_2$ on these words is equal to $|v|+1$. Therefore,

$$\#[q_2, L \cap \Sigma^n] = \sum_{i=0}^{n-2} 2^i \sum_{m>0} \binom{n-i-2-m}{m}(i+1).$$

Similarly the load of $q_1$ on the words of above form is equal to $m$, so it holds that

$$\#[q_2, L \cap \Sigma^n] = \sum_{i=0}^{n-2} 2^i \sum_{m>0} \binom{n-i-2-m}{m}(m+1).$$

From the fact that

$$\sum_{m>0} \binom{n-i-2-m}{m} = F_{n-i-1},$$

where $F_n$ is the $n$-th Fibonacci number, we obtain

$$\#[q_2, L \cap \Sigma^n] = \sum_{i=0}^{n-2} 2^i (i+1) F_{n-i-1},$$

and

$$\#[q_1, L \cap \Sigma^n] \le \sum_{i=0}^{n-2} 2^i \sum_{m>0} \binom{n-i-2-m}{m}(n-i-2) = \sum_{i=0}^{n-2} 2^i (n-i-2) F_{n-i-1}.$$

It is a well known fact that there are constants $a, b$ such that $a\phi^n < F_n < b\phi^n$, where

*Theorem 4.3 is proved independently of the result $\mathcal{L}_{\text{EQA}} \subsetneq \mathcal{R}$.

$\phi$ is the golden ratio. From that we can compute a lower bound of $\#[q_2, L \cap \Sigma^n]$ as

$$\#[q_2, L \cap \Sigma^n] \geq \sum_{i=0}^{n-2} 2^i (i+1) a\phi^n - i - 2 = a\phi^{n-2} \cdot \sum_{i=0}^{n-2} (i+1) \left(\frac{2}{\phi}\right)^i.$$

By simplifying the sum we obtain

$$\#[q_2, L \cap \Sigma^n] \geq c_2 \phi^{n-2} (n-2) \left(\frac{2}{\phi}\right)^{n-1}.$$

The analogous computation can be done for the load of the state $q_1$:

$$\begin{aligned}
\#[q_1, L \cap \Sigma^n] &\leq b\phi^{n-2} \sum_{i=0}^{n-2} (n-i-2) \left(\frac{2}{\phi}\right)^i \\
&\leq b\phi^{n-2} \sum_{j=0}^{n-3} \sum_{i=0}^{j} \left(\frac{2}{\phi}\right)^i \\
&\leq \frac{b}{(\frac{2}{\phi} - 1)} \phi^{n-2} \sum_{j=0}^{n-3} \left(\frac{2}{\phi}\right)^i \\
&\leq c_1 \phi^{n-2} \left(\frac{2}{\phi}\right)^{n-2} \leq c_1 \phi^{n-2} \left(\frac{2}{\phi}\right)^{n-1}.
\end{aligned}$$

From above we get that $\#[q_2, L \cap \Sigma^n] > ((n-2)c_2/c_1) \cdot \#[q_1, L \cap \Sigma^n]$, therefore for any $\ell$ from Theorem 4.3 there exists an arbitrarily large number $n$ such that

$$\#[q_2, L \cap \Sigma^n] > \ell \#[q_1, L \cap \Sigma^n].$$

Hence, $B$ is not equiloaded and $\mathcal{L}_{\text{EQA}} \neq \mathcal{R}$. □

**Corollary 4.1.** Every finite language is in $\mathcal{L}_{\text{EQA}}$.

*Proof.* Follows directly from Theorem 4.1. □

## 4.3 Relationship with Minimal-State Automaton

Unlike in the case of strict equiloadedness, there is an language $L \in \mathcal{L}_{\text{EQA}}$ such that the minimal-state automaton for $L$ is not equiloaded. Therefore, it is harder to show that some language $L$ does not belong to $\mathcal{L}_{\text{EQA}}$ because it does not suffice to prove that the minimal-state automaton is not equiloaded. However, there is a relationship between minimal-state automaton and equiloadedness of a language, as we shall see in Theorem 4.3

**Theorem 4.2.** There exists a language $L \in \mathcal{L}_{\text{EQA}}$ such that the minimal-state automaton for the language $L$ is not equiloaded.


(a) The minimal-state automaton (b) The equiloaded automaton

Figure 4.3: The minimal-state automaton and the equiloaded automaton for the language $L = \{ab, ba\}^*$

*Proof.* Consider the language $L = \{\mathrm{ab}, \mathrm{ba}\}^*$. The minimal state automaton in Figure 4.3(a) for $L$ is $A_1 = (\{q_0, q_1, q_2\}, \{\mathrm{a}, \mathrm{b}\}, \delta_1, q_0, \{q_0\})$, where $\delta_1$ is defined by

$$\delta_1(q_0, \mathrm{a}) = q_1, \quad \delta_1(q_0, \mathrm{b}) = q_2,$$
$$\delta_1(q_1, \mathrm{b}) = q_0, \quad \delta_1(q_2, \mathrm{a}) = q_0.$$

We can see that $A_1$ is not equiloaded, because

$$\sum_{w \in L \cap \Sigma^{2n}} \#[q_0, w] = (n+1) \cdot 2^n,$$
$$\frac{(2n+1) \cdot 2^n}{3} = \frac{2n+1}{3} \cdot 2^n,$$

and from this and Lemma 4.1 we get $k \geq (n-2)/3$, so $k$ is not a constant.

Now it suffices to show that $L \in \mathcal{L}_{\mathrm{EQA}}$. In order to find an equiloaded automaton for $L$, we can look at the automaton $A_1$ and determine that $q_0$ is used every time $q_1$ or $q_2$ is used. So we can try to split the state $q_0$ into two states. We obtain an automaton in Figure 4.3(b) $A_2 = (\{q_0, q_1, q_2, q_3\}, \{\mathrm{a}, \mathrm{b}\}, \delta_2, q_0, \{q_0, q_3\})$, where $\delta_2$ is defined by

$$\delta_2(q_0, \mathrm{a}) = q_1, \quad \delta_2(q_0, \mathrm{b}) = q_2, \quad \delta_2(q_1, \mathrm{b}) = q_3,$$
$$\delta_2(q_2, \mathrm{a}) = q_0, \quad \delta_2(q_3, \mathrm{a}) = q_1, \quad \delta_2(q_3, \mathrm{b}) = q_2.$$

This automaton is equiloaded. One way to prove this is by using algebraic representation from Section 4.7. Another way is to directly use the definition, or the equivalent form from Lemma 4.1. It is easy to see that the most used state will be $q_0$ or $q_3$ because it holds that $\#[q_0, w] \geq \#[q_2, w]$ and $\#[q_3, w] \geq \#[q_1, w]$. It is easy to see that, for a word $w$ consisting of $i$ pairs "ab" and $j$ pairs "ba", the load of $q_0$ is $\#[q_0, w] = j + 1$ and the load of $q_1$ is $\#[q_3, w] = i$. Therefore, it holds that

$$\#[q_0, L \cup \Sigma^{2n}] = \sum_{i=0}^{n} \binom{n}{i} (i+1) = 2^{n-1}(n+2)$$
$$\#[q_3, L \cup \Sigma^{2n}] = \sum_{i=0}^{n} \binom{n}{i} (i) = 2^{n-1} n.$$

The most used state is $q_0$. The average load of states is $(2n+1)2^{n-2}$, therefore there exists $k$ such that

$$2^{n-1}(n+2) - (2n+1)2^{n-2} = 3 \cdot 2^{n-2} \leq k|L \cap \Sigma^{2n}| = k2^n.$$

Thus $A_2$ is equiloaded and $L \in \mathcal{L}_{\text{EQA}}$. □

**Theorem 4.3.** Let $L$ be a language. If for the minimal-state automaton $A$ for the language $L$ it holds that

$$\exists p, q \in Q\ \forall \ell \in \mathbb{R}^+\ \forall n_0 \in \mathbb{N}\ \exists n \in \mathbb{N},\ n \geq n_0 \quad \#[p, L \cap \Sigma^n] > \ell \#[q, L \cap \Sigma^n],$$

then $L \notin \mathcal{L}_{\text{EQA}}$.

*Proof.* Suppose that $L \in \mathcal{L}_{\text{EQA}}$ and let $B = (Q_B, \Sigma, \delta_B, q_{0B}, F_B)$ be an arbitrary equiloaded automaton such that $L = L(B)$. Let $k_B$ be the equiloadedness constant of $B$. From the condition in the theorem, there are states $p, q$ such that the inequality holds. Some of states from $Q_B$ (let us call them $p_1, \ldots, p_c$) compose the equivalence class from Myhill-Nerode Theorem, which belongs to the state $p$. Similarly, there are some states $q_1, \ldots, q_d$, which belong to the state $q$. At least for one of the states $p_1, \ldots, p_c$ (let it be $p_1$) it holds

$$\#_B[p_1, L \cap \Sigma^n] \geq \frac{\#_A[p, L \cap \Sigma^n]}{c}.$$

Similarly, without loss of generality, suppose that for $q_1$ it holds

$$\#_B[q_1, L \cap \Sigma^n] \leq \frac{\#_A[q, L \cap \Sigma^n]}{d}.$$

Let $\ell = (k_B + 1)c$. Then

$$\#[p_1, L \cap \Sigma^n] \geq \frac{\#[p, L \cap \Sigma^n]}{c} > \frac{(k_B+1)d\#[q, L \cap \Sigma^n]}{d} \geq (k_B+1)d\#[q_1, L \cap \Sigma^n].$$

From the above inequality we have

$$\#[p_1, L \cap \Sigma^n] - \#[q_1, L \cap \Sigma^n] > k_B d \#[q_1, L \cap \Sigma^n].$$

If $\#[q_1, L \cap \Sigma^n] > |L \cap \Sigma^n|/d$, it is clear that $B$ cannot be equiloaded, thus we can assume the contrary. But then $\#[q_1, L \cap \Sigma^n] \leq |L \cap \Sigma^n|$ and there is some state, that is used more or equal to the average load of all states, thus

$$\frac{(n+1)|L \cap \Sigma^n|}{|Q|} - |L \cap \Sigma^n| = \frac{(n+1-|Q|)|L \cap \Sigma^n|}{|Q|},$$

so $B$ is not equiloaded. Therefore, our first assumption $L \in \mathcal{L}_{\text{EQA}}$ must be false. □

## 4.4 Automata Transformations

**Theorem 4.4.** Let $A$ be a DFA. Let $B$ be a DFA, obtained from $A$ by a sequence of arbitrary changes in the transition function from $\delta(p,x)=q$ to $\delta(p,y)=q$ in such a way that $B$ is still a deterministic finite automaton. If $A$ is equiloaded, then $B$ is equiloaded.

*Proof.* The graphical representation of the automaton $A$ is equal to the graphical representation of the automaton $B$. Thus, for every word $w \in L(A)$ there is exactly one word $w' \in L(B)$ such that $B$ uses the same sequence of transitions on $w'$ as $A$ uses on $w$. A load of a particular state $q$ in the automaton $A$ on a word $w$ will be the same as the load of $q$ in the automaton $B$ on a word $w'$. Therefore $\#_A[q, L(A) \cap \Sigma^n] = \#_B[q, L(B) \cap \Sigma^n]$, so $B$ is equiloaded. □

**Definition 4.2.** Let $A = (Q_A, \Sigma_A, \delta_A, q_{0A}, F_A)$, $B = (Q_B, \Sigma_B, \delta_B, q_{0B}, F_B)$ be deterministic finite automata such that $\Sigma_A \cap \Sigma_B = \emptyset$ . The *$A, B$-composition automaton* $C = (Q_A \times Q_B, \Sigma_A \cup \Sigma_B, \delta_C, (q_{0A}, q_{0B}), F_A \times F_B)$ is a DFA with $\delta_C$ such that

$$\begin{array}{lll} \forall x \in \Sigma_B & \delta_C((q_A, q_B), x) = (q_A, p_B) \stackrel{\text{def}}{\Longleftrightarrow} \delta_B(q_B, x) = p_B, \\ \forall x \in \Sigma_A & \delta_C((q_A, q_B), x) = (p_A, q_{0B}) \stackrel{\text{def}}{\Longleftrightarrow} \delta_A(q_A, x) = p_A \text{ and } q_B \in F_B. \end{array}$$

We shall write $C = A \circ B$.

*Example* 4.2. Let $A$ be a minimal-state automaton for a language $L_1 = \{\mathrm{a}^{3n+2} \mid n \in \mathbb{N}\}$ and let $B$ be a minimal-state automaton for $L_2 = (\{\mathrm{b}\} \cdot \{\mathrm{b}, \mathrm{cc}\})^* \cdot \{\mathrm{b}\}$. (See figure 4.4.) Then automaton $C = A \circ B$ is an automaton that accepts a language

$$L_3 = \{w_1 \mathrm{a} w_2 \dots \mathrm{a} w_{3n} \mid n \in \mathbb{N}, \forall i \leq 3n \ w_i \in L_2\}.$$

**Theorem 4.5.** Let $A = (Q_A, \Sigma_A, \delta_A, q_{0A}, F_A)$, $B = (Q_B, \Sigma_B, \delta_B, q_{0B}, F_B)$ be equiloaded automata such that $\Sigma_A \cap \Sigma_B = \emptyset$, and the equiloadedness constants for both $A$ and $B$ are equal to 0. Then the automaton $A \circ B$ is equiloaded with equiloadedness constant 0.

*Proof.* We compute the load of each state while the automaton $C = A \circ B$ processes all words of length $n$. It can be shown that a word $w \in L(C) = L$ of the length $n$ has the form $w = w_1 a_1 w_2 a_2 \dots w_k$, where $w_i \in L(B)$, $a_1 a_2 \dots a_{n-1} \in L(A)$. We denote $L \cap \Sigma^n$ by $L^{(n)}$. We shall show that

$$\begin{aligned} |\#_C[(p, q_1), L^{(n)}] - \#_C[(p, q_2), L^{(n)}]| &\leq 0 \text{ and} \\ |\#_C[(p_1, q), L^{(n)}] - \#_C[p_2, q, L^{(n)}]| &\leq 0. \end{aligned}$$

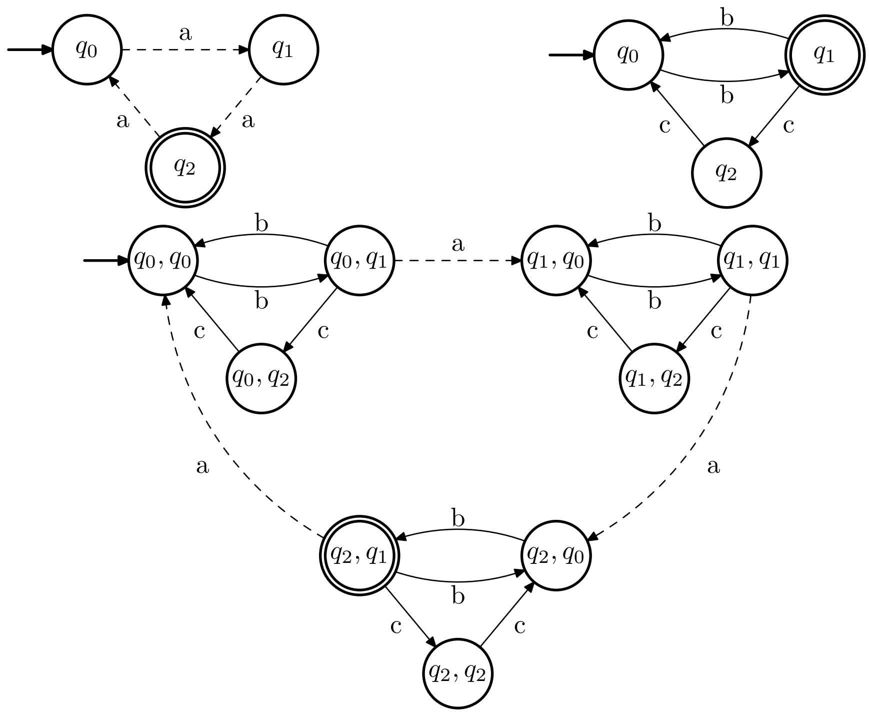



Figure 4.4: The automata $A$ (left) and $B$ (right) compose the automaton $C = A \circ B$ (below).

Then it holds that

$$\begin{aligned}
& |\#_C[(p_1, q_1), L^{(n)}] - \#_C[(p_2, q_2), L^{(n)}]| \\
= \; & |\#_C[(p_1, q_1), L^{(n)}] - \#_C[(p_1, q_2), L^{(n)}] + \#_C[(p_1, q_2), L^{(n)}] - \#_C[(p_2, q_2), L^{(n)}]| \\
\leq \; & |\#_C[(p_1, q_1), L^{(n)}] - \#_C[(p_1, q_2), L^{(n)}]| + |\#_C[(p_1, q_2), L^{(n)}] - \#_C[(p_2, q_2), L^{(n)}]| \\
\leq \; & 0,
\end{aligned}$$

and $A \circ B$ is equiloaded.

It is clear from the construction of $A \circ B$ that

$$L(A \circ B) = \{w_1 a_1 w_2 \dots a_{n-1} w_n \mid w_i \in L(B),\ a_1 a_2 \dots a_n \in L(A)\}.$$

Therefore, for all words of a given length $n$, it holds that

$$L^{(n)} = \bigcup_{i=0}^{n} L_i^{(n)},$$

where

$$L_i^{(n)} = \{w_1 a_1 \dots a_i w_{i+1} \mid w_k \in L(B),\ a_1 \dots a_i \in L(B),\ \sum_{k=1}^{i+1} |w_k| = n - i\}.$$

Firstly, we shall compute $|L^{(n)}|$. It holds that

$$|L^{(n)}| = \sum_{i=0}^{n} |L_i^{(n)}| = \sum_{i=0}^{n} |L(A)^{(i)}| \cdot \sum_{j=0}^{n-i} \binom{n-j-1}{i-1} (i+1)|L(B)^{(j)}|,$$

because for given $a_1, \ldots, a_i$, there are $(n-j-1)$ choose $(i-1)$ words in $L^{(i)}$ such that the first substring from $L(B)$ has the length $j$.

Now we shall compute $\#[(p,q), L^{(n)}]$ by summing the load of $(p,q)$ on all words from $L_i^{(n)}$. Similarly as in the case of counting words from $L_i^{(n)}$ it holds that

$$\begin{aligned}
\#[(p,q), L^{(n)}] &= \sum_{i=0}^{n} \#[p, L(A)^{(i)}] \cdot \sum_{j=0}^{n-i} \binom{n-j-1}{i-1} (i+1)\#[q, L(B)^{(j)}] \\
&= \sum_{i=0}^{n} \frac{(i+1)|L(A)^{(i)}|}{|Q_A|} \cdot \sum_{j=0}^{n-i} \binom{n-j-1}{i-1} (i+1) \frac{(j+1)|L(B)^{(j)}|}{|Q_B|}.
\end{aligned}$$

In the above equation we can see that the load of a state $(p,q)$ does not depend on $p$ or $q$, therefore the load is the same for $(p,q)$ and $(p,q')$, or for $(p,q)$ and $(p',q)$. It means that any two states are equally loaded on words from $L^{(n)}$, therefore equiloadedness constant for $C$ is equal to 0. □

The previous theorem does not hold if the equiloadedness constant for $A$ or $B$ is non-zero.

**Theorem 4.6.** Let $\Sigma_A$, $\Sigma_B$ be finite alphabets such that $\Sigma_A \cap \Sigma_B = \emptyset$. There exist equiloaded automata $A = (Q_A, \Sigma_A, \delta_A, q_{0A}, F_A)$, $B = (Q_B, \Sigma_B, \delta_B, q_{0B}, F_B)$ such that $A \circ B$ is not equiloaded.

*Proof.* Let $A$ be an automaton for the language $L_A = \{\mathrm{a}, \mathrm{b}\}$ and $B$ be the minimal-state automaton for the language $L_B = \{\mathrm{c}^i \mid i \in \mathbb{N}\}$. A language accepted by $A \circ B$ is the language $L = \{\mathrm{c}^i x \mathrm{c}^j \mid i, j \in \mathbb{N}, x \in \{\mathrm{a}, \mathrm{b}\}\}$. The automaton $A \circ B$ is not equiloaded, although the language $L \in \mathcal{L}_{\mathrm{EQA}}$. The load of the state $(q_{0A}, q_{0B})$ is

$$\#[(q_{0A}, q_{0B}), L \cap \Sigma^n] = 2 \sum_{i=0}^{n-1} i + 1 = n(n+1).$$

The average load of the states of $A \circ B$ is $(n+1)n/3$. Thus, there is no constant $k$ such that

$$n(n+1) - \frac{n(n+1)}{3} = \frac{2n(n+1)}{3} \leq k(n).$$

□

## 4.5 Closure Properties of $\mathcal{L}_{\text{EQA}}$

In this section we shall prove several theorems about closure properties of the family $\mathcal{L}_{\text{EQA}}$. Most of them are proved using Theorem 4.3.

**Theorem 4.7.** The family $\mathcal{L}_{\text{EQA}}$ is not closed under union.

*Proof.* Consider languages $L_{\text{a}} = \{\text{a}^i \mid i \in \mathbb{N}\}$ and $L_{\text{b}} = \{\text{b}^i \mid i \in \mathbb{N}\}$. It is clear that $L_{\text{a}}, L_{\text{b}} \in \mathcal{L}_{\text{EQA}}$. In an automaton $A$ which accepts $L = L_{\text{a}} \cup L_{\text{b}}$, there must be an initial state $q_0$ such that automaton $A$ will not return to $q_0$ during computation. Therefore, the load of $q_0$ on words of a given length $n$ is equal to the number of words of the length $n$. (Which is equal to 2: $\text{a}^n$ and $\text{b}^n$.) Moreover, there is a state $q \in Q_A$ such that

$$\#[q, L \cap \Sigma^n| \geq \frac{(n+1)|L \cap \Sigma^n|}{|Q_A|} = \frac{2(n+1)}{|Q_A|}.$$

From the above we get

$$\#[q, L \cap \Sigma^n] - \#[q_0, L \cap \Sigma^n] \geq \frac{2(n+1)}{|Q_A|} - 2 = \left(\frac{n+1}{|Q_A|} - 1\right)|L \cap \Sigma^n|,$$

thus, $A$ is not equiloaded. □

**Theorem 4.8.** The family $\mathcal{L}_{\text{EQA}}$ is not closed under homomorphism.

*Proof.* Let $L = \{\text{a}^i\text{ba}^j \mid i,j \in \mathbb{N}\}$. This language is in the family $\mathcal{L}_{\text{EQA}}$, we can construct the automaton for the $L$ by relabelling one arc in the graphical representation of the minimal state automaton for the language $L_1 = \{\text{a}^i\text{b}^j \mid i,j \in \mathbb{N},\ j \neq 0\}$. (We relabel the loop $(q_1, q_1)$ by a.) Let $h$ be a homomorphism such that $h(\text{a}) = \text{a}$, $h(\text{b}) = \text{bb}$. Then the minimal-state automaton $A = (\{q_0, q_1, q_2\}, \{\text{a}, \text{b}\}, \delta_A, q_0, \{q_2\})$ for the language $h(L) = \{\text{a}^i\text{bba}^j \mid i,j \in \mathbb{N}\}$ fulfils the condition from Theorem 4.3, so $h(L) \notin \mathcal{L}_{\text{EQA}}$. (The transition function $\delta_A$ is defined by $\delta_A(q_0, \text{a}) = q_0$, $\delta_A(q_0, \text{b}) = q_1$, $\delta_A(q_1, \text{a}) = q_1$). For showing it we compare the load of states $q_1$ and $q_2$ in the automaton $A$.

On the word $\text{a}^i\text{bba}^j \in h(L)$ the state $q_1$ is used once, and the state $q_2$ is used $j+1$ times. Because each word is determined by position of the first occurrence of the symbol b, the number of words of a given length $n$ is $n$. Therefore, we obtain

$$\#[q_2, h(L) \cap \Sigma^n] = \sum_{i=0}^{n-1}(n-i) = \frac{n^2+n}{2},$$

and

$$\#[q_1, h(L) \cap \Sigma^n] = \sum_{i=0}^{n-1} 1 = n.$$

Thus, for any $\ell$ there exists arbitrarily large $n$ such that the load of the state $q_2$ is at least $\ell$ times greater than the load of the state $q_1$. This ensures that we can use Theorem 4.3, hence $h(L) \notin \mathcal{L}_{\text{EQA}}$. □

**Theorem 4.9.** The family $\mathcal{L}_{\mathrm{EQA}}$ is not closed under concatenation.

*Proof.* Consider languages $L_1 = \{\mathrm{a}^i \mid i \in \mathbb{N}\}$ and $L_2 = \{\mathrm{b}\}$. They both belong to the family $\mathcal{L}_{\mathrm{EQA}}$, for the first the minimal-state automaton is equiloaded, the second is finite thus belongs to $\mathcal{L}_{\mathrm{SEQA}} \subseteq \mathcal{L}_{\mathrm{EQA}}$. We will show that the concatenation of these languages $L = L_1 \cdot L_2 = \{\mathrm{a}^i\mathrm{b} \mid i \in \mathbb{N}\}$ is not in the family $\mathcal{L}_{\mathrm{EQA}}$, because for the minimal-state automaton $A = (\{q_0, q_1\}, \Sigma, \delta, q_0, \{q_1\})$ the condition from Theorem 4.3 holds: $(L \cap \Sigma^n = \{\mathrm{a}^{n-1}\mathrm{b}\})$

$$\#[q_0, L \cap \Sigma^n] = n, \quad \#[q_1, L \cap \Sigma^n] = 1,$$

so for a number $\ell$ there is a length $n = \ell + 1$ such that the load of $q_0$ is greater than $\ell$ times the load of $q_1$. □

**Theorem 4.10.** The family $\mathcal{L}_{\mathrm{EQA}}$ is not closed under intersection.

*Proof.* As we showed in the proof of Theorem 4.11, the deterministic finite automaton $A = (\{q_0, q_1, q_2\}, \{\mathrm{a}, \mathrm{b}\}, \delta, q_0, \{q_2\})$, where the transition function is defined by

$$\begin{aligned} &\delta(q_0, \mathrm{a}) = q_0, \quad \delta(q_0, \mathrm{b}) = q_1, \quad \delta(q_1, \mathrm{c}) = q_1, \\ &\delta(q_1, \mathrm{b}) = q_2, \quad \delta(q_2, \mathrm{a}) = q_2, \end{aligned}$$

is equiloaded. It is easy to see that the language accepted by the automaton $A$ is $L = \{\mathrm{a}^i\mathrm{b}\mathrm{c}^j\mathrm{b}\mathrm{a}^k \mid i, j, k \in \mathbb{N}\}$. By relabeling the loop $(q_1, q_1)$ by $d$ we obtain equiloaded automaton $A'$ for the language $L' = \{\mathrm{a}^i\mathrm{b}\mathrm{d}^j\mathrm{b}\mathrm{a}^k \mid i, j, k \in \mathbb{N}\}$. The intersection of $L$ and $L'$ is $L_1 \cap L_2 = \{\mathrm{a}^i\mathrm{b}\mathrm{b}\mathrm{a}^k \mid i, k \in \mathbb{N}\}$. This language is not equiloaded, as was shown in the proof of Theorem 4.8. Hence, the family $\mathcal{L}_{\mathrm{EQA}}$ is not closed under intersection. □

**Theorem 4.11.** The family $\mathcal{L}_{\mathrm{EQA}}$ is not closed under inverse homomorphism.

*Proof.* Consider a language $L = \{\mathrm{a}^i\mathrm{b}\mathrm{c}^j\mathrm{b}\mathrm{a}^k \mid i, j, k \in \mathbb{N}\}$ and homomorphism $h$ such that $h(\mathrm{a}) = \mathrm{a}$, $h(\mathrm{b}) = \mathrm{b}$, $h(\mathrm{c}) = \mathrm{a}$. Then $L' = h^{-1}(L) = \{\mathrm{a}^i\mathrm{b}\mathrm{b}\mathrm{a}^j \mid i, j \in \mathbb{N}\}$. We already know from the proof of Theorem 4.8 that $L' \notin \mathcal{L}_{\mathrm{EQA}}$. We will show that $L \in \mathcal{L}_{\mathrm{EQA}}$. Consider the minimal-state automaton $A = (\{q_0, q_1, q_2\}, \{\mathrm{a}, \mathrm{b}\}, \delta, q_0, \{q_2\})$ for language $L$, where the transition function is defined by

$$\begin{aligned} &\delta(q_0, \mathrm{a}) = q_0, \quad \delta(q_0, \mathrm{b}) = q_1, \quad \delta(q_1, \mathrm{c}) = q_1, \\ &\delta(q_1, \mathrm{b}) = q_2, \quad \delta(q_2, \mathrm{a}) = q_2. \end{aligned}$$

Consider a word $\mathrm{a}^i\mathrm{b}\mathrm{c}^j\mathrm{b}\mathrm{a}^k$ of length $n$. On this word, the load of the state $q_0$ is equal to $i + 1$, the load of the state $q_1$ is equal to $j + 1$, and the load of the state $q_2$ is equal to $k + 1 = n - i - j - 1$. The number of words of length $n$ is equal to $(n^2 + n - 2)/2$

Let us compute the load on words from $L \cap \Sigma^n$:

$$
\begin{aligned}
\#[q_0, L \cap \Sigma^n] &= \sum_{i=0}^{n-2} \sum_{j=0}^{n-i-1} i+1 = \sum_{i=0}^{n-2} (i+1)(n-i) \\
&= (n-1)\left(\frac{(n-1)(n-2)}{2}\right) + n(n-1) - \frac{(n-2)(n-1)(2n-3)}{6} \\
&= \frac{n(n-1)(n+4)}{6} = \frac{n^3+3n^2-4n}{2}, \\
\#[q_1, L \cap \Sigma^n] &= \sum_{i=0}^{n-2} \sum_{j=0}^{n-i-1} j+1 = \sum_{i=0}^{n-2} \frac{(n-i)(n-i+1)}{2} \\
&= \frac{(n-1)(n^2+4n+6)}{6} = \frac{n^3+3n^2+2n-6}{6}, \\
\#[q_2, L \cap \Sigma^n] &= \sum_{i=0}^{n-2} \sum_{j=0}^{n-i-1} n-i-j-1 = \sum_{i=0}^{n-2} \frac{(n-i)(n-i-1)}{6} \\
&= \frac{(n-1)n(n+1)}{6} = \frac{n^3-n}{6}.
\end{aligned}
$$

Hence we have

$$
\begin{aligned}
|\#[p, L \cap \Sigma^n] - \#[q, L \cap \Sigma^n]| &\le \frac{(n^3+3n^2+2n-6)-(n^3-n)}{6} \\
&= \frac{3n^2+3n-6}{6} = |L \cap \Sigma^n|.
\end{aligned}
$$

Thus $A$ is equiloaded, $L \in \mathcal{L}_{\mathrm{EQA}}$, $h^{-1}(L) = L' \notin \mathcal{L}_{\mathrm{EQA}}$, hence the family $\mathcal{L}_{\mathrm{EQA}}$ is not closed under inverse homomorphism. □

**Theorem 4.12.** The family $\mathcal{L}_{\mathrm{EQA}}$ is not closed under reversal.

*Proof.* Consider the language $L = \{\mathrm{a}^i\mathrm{b}^j \mid i, j \in \mathbb{N}\}$. We will show that $L \in \mathcal{L}_{\mathrm{EQA}}$, but $L^C \notin \mathcal{L}_{\mathrm{EQA}}$. The minimal-state automaton $A = (\{q_0, q_1\}, \Sigma, \delta_A, q_0, \{q_0, q_1\})$ for the language $L$, where the transition function is defined by

$$\delta_A(q_0, \mathrm{a}) = q_0, \quad \delta_A(q_0, \mathrm{b}) = q_1, \quad \delta_A(q_1, \mathrm{b}) = q_1,$$

is equiloaded. To see it, let us compute the load of state $q_0$ and $q_1$ respectively.

$$
\begin{aligned}
\#[q_0, L \cap \Sigma^n] &= \sum_{i=0}^{n-1} i+1 = \frac{i^2+i}{2}, \\
\#[q_1, L \cap \Sigma^n] &= \sum_{i=0}^{n-1} n-i = \frac{i^2+i}{2}.
\end{aligned}
$$

It means that $A$ is equiloaded with the equiloadedness constant equal to 0.

The minimal-state automaton $B = (\{q_0, q_1, q_2\}, \Sigma, \delta_B, q_0, \{q_1, q_2\})$ for language $L^R$

has the transition function defined by

$$\delta_B(q_0, \mathrm{b}) = q_1, \quad \delta_B(q_1, \mathrm{b}) = q_1, \quad \delta_B(q_1, \mathrm{a}) = q_2, \quad \delta_B(q_2, \mathrm{a}) = q_2.$$

It is obvious that the state $q_0$ is used exactly once on every word, thus $|L \cap \Sigma^n|$ times on all words of a given length $n$. For length $n$ there exists an state $q$ used at least $(n+1)|L \cap \Sigma^n|/3$ times. This means that for any $\ell$ there is an arbitrary big $n$ such that $\#[q, L \cap \Sigma^n]$ is greater than $\ell \cdot \#[q_0, L \cap \Sigma^n]$. Therefore by Theorem 4.3 $L^R \notin \mathcal{L}_{\mathrm{EQA}}$ and thus $\mathcal{L}_{\mathrm{EQA}}$ is not closed under reversal.

□

**Theorem 4.13.** The family $\mathcal{L}_{\mathrm{EQA}}$ is not closed under complement.

*Proof.* Every finite language belongs to the family $\mathcal{L}_{\mathrm{EQA}}$, so does $L = \{\mathrm{aa}\} \in \mathcal{L}_{\mathrm{EQA}}$. We will show that the minimal-state automaton $A = (\{q_0, \ldots, q_3\}, \{\mathrm{a}, \mathrm{b}\}, \delta, q_0, \{q_0, q_1, q_3\})$ shown in Figure 4.5 fulfils the condition from Theorem 4.3. On every word $w$ the state $q_0$ is used only once and the state $q_3$ is used at least $|w| - 2$ times. Therefore, it is easy to see that for any $\ell$ there exists arbitrarily large $n$ ($n > \ell + 3$) such that the load of the state $q_2$ is greater than $\ell$ times the load of the state $q_0$. The language $L^R$ is not in the family $\mathcal{L}_{\mathrm{EQA}}$, thus $\mathcal{L}_{\mathrm{EQA}}$ is not closed under complement. □

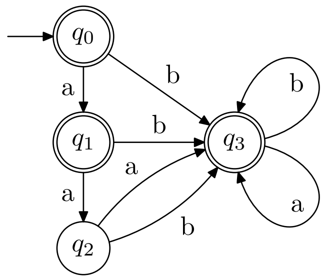



Figure 4.5: The minimal-state automaton for the language $\{\mathrm{a}, \mathrm{b}\}^* - \{\mathrm{aa}\}$.

We left open the problem of determining whether the family $\mathcal{L}_{\mathrm{SEQA}}$ is closed under Kleene star and Kleene plus, we believe that the answer is negative in both cases. The reader can compare our results with closure properties of the family of regular languages in Table 4.1.

| | $\cup$ | $\cap$ | $\cdot$ | $h$ | $h^{-1}$ | $C$ | $R$ | $*$ | $+$ |
|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{R}$ | yes | yes | yes | yes | yes | yes | yes | yes | yes |
| $\mathcal{L}_{\mathrm{EQA}}$ | no | no | no | no | no | no | no | open | open |
| $\mathcal{L}_{\mathrm{SEQA}}$ | no | yes | no | no | no | no | no | no | no |

Table 4.1: Closure properties of $\mathcal{R}$, $\mathcal{L}_{\mathrm{EQA}}$ and $\mathcal{L}_{\mathrm{SEQA}}$

## 4.6 Characterization

Since it is not true that $L \in \mathcal{L}_{\text{EQA}}$ if and only if the minimal-state automaton for $L$ is equiloaded, it is harder to characterize the family $\mathcal{L}_{\text{EQA}}$. In this section, we present our approaches to the characterization of the family $\mathcal{L}_{\text{EQA}}$, although we will not prove this characterization. We shall state some theorems that partially characterize this family, but a final characterization is still open.

The characterization of the family $\mathcal{L}_{\text{SEQA}}$ was built on the graphical representation of automata regardless of a set of accepting states $F$. However, in the case of the family $\mathcal{L}_{\text{EQA}}$ we shall see that graphical representation of automata alone does not suffice.

There are two ways to create a partial characterization. One way is to formulate a condition such that if it holds for an automaton $A$, then $A$ is not equiloaded (or, conversely, if $A$ is equiloaded, then condition is not true). A good candidate for this condition seems to be a bridge in the graphical representation of an automaton, but as the theorem below demonstrates, this condition is not really good for our purposes.

**Theorem 4.14.** There is an equiloaded automaton $A$ such that the graphical representation of $A$ contains a bridge.

*Proof.* Such an automaton is, for example, any automaton for a finite language, or the minimal-state automaton for the language $L = \{\mathrm{a}^i\mathrm{b}^j \mid i, j \in \mathbb{N}\}$ considered in the proof of Theorem 4.1. □

Another type of condition which will partially characterize the family $\mathcal{L}_{\text{SEQA}}$ is such a condition that if it holds, then $A$ is equiloaded. A good candidate for this type of condition seems to be "equal number of cycles through each state." But, unfortunately, this is not a sufficient condition.

**Theorem 4.15.** There is a non-equiloaded automaton $A$ such that there is an equal number of cycles through each state.

*Proof.* Consider the minimal-state automaton $A = (\{q_0, q_1, q_2\}, \{\mathrm{a}, \mathrm{b}\}, \delta, q_0, \{q_1\})$ for the language $L = \{\mathrm{a}^i\mathrm{b}^{2j+1} \mid i, j \in \mathbb{N}\}$ that has one cycle through the state $q_0$ and one cycle through the states $q_1$, $q_2$. Therefore, there is an equal number of cycles through all states. This automaton is not equiloaded. □

## 4.7 Algebraic Representation

We shall now introduce an algebraic representation for DFA. This representation can be used to show (non-)equiloadedness of automata.

Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA. For $n \in \mathbb{N}$, we shall use

$$M_n = \begin{pmatrix} m^n_{0,0} & m^n_{0,1} & \cdots & m^n_{0,|Q|-1} \\ m^n_{1,0} & m^n_{1,1} & \cdots & m^n_{1,|Q|-1} \\ \vdots & \vdots & \ddots & \vdots \\ m^n_{|Q|-1,0} & m^n_{|Q|-1,0} & \cdots & m^n_{|Q|-1,|Q|-1} \end{pmatrix}, \quad \alpha_n = \begin{pmatrix} \alpha^n_0 \\ \alpha^n_1 \\ \vdots \\ \alpha^n_{|Q|-1} \end{pmatrix},$$

where $M_n$ is a matrix (of the size $|Q| \times |Q|$) where the element $m^n_{i,j}$ equals to the load of the state $q_j$ at all words of length $n$, for which the computation of $A$ finishes in the state $q_i$. The $\alpha_n$ is a column vector such that $\alpha^n_i$ is the number of words of length $n$, for which $A$'s computation finishes in the state $q_i$.

Without loss of generality, for $n = 0$ it holds that

$$M_0 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}, \quad \alpha_0 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

If we want to compute $M_n$ from $M_{n-1}$ (and $\alpha_{n-1}$), we need a transition matrix $\Delta$ – from the transition function of $A$. The element $\delta_{i,j}$ is the number of transitions from $q_j$ to $q_i$. (For a better insight see the proof of Theorem 4.16.) Now we can create formulae for both $\alpha_n$ and $A_n$:

$$\alpha_n = \Delta \cdot \alpha_{n-1}, \tag{4.2}$$

$$M_n = \Delta \cdot M_{n-1} + \mathrm{diag}(\alpha_n). \tag{4.3}$$

*Remark.* The function *diag* is defined by

$$\mathrm{diag}\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{pmatrix} = \begin{pmatrix} a_0 & 0 & \cdots & 0 \\ 0 & a_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_k \end{pmatrix}.$$

If we are able to compute the matrix $M_n$, we can easily compute the minimal value of the constant $k$ from Definition 4.1 such that the condition from this definition holds for $n$. We can construct a vector $\varphi$ with 1 at the $i$-th element if $q_i$ is an accepting state and 0 otherwise. By multiplying $\varphi$ and $M_n$ we obtain a vector $\psi$ with the load of the state $q_i$ at the $i$-th element of the vector $\psi$ (on words from $L(A)$ of length $n$). The maximal element of $\psi$ contains the load of the maximally used state. We can compute the average load by summing all elements of $\psi$ and dividing by $|Q|$.

Now we will show an example of proving the equiloadedness of an automaton using the algebraic representation.

**Theorem 4.16.** An automaton $A = (\{q_0, q_1, q_2, q_3\}, \{\mathrm{a}, \mathrm{b}\}, \delta, q_0, \{q_0, q_3\})$, where $\delta$ is defined by

$$\delta(q_0, \mathrm{a}) = q_1, \quad \delta(q_0, \mathrm{b}) = q_3, \quad \delta(q_1, \mathrm{b}) = q_2,$$
$$\delta(q_3, \mathrm{a}) = q_0, \quad \delta(q_2, \mathrm{a}) = q_1, \quad \delta(q_2, \mathrm{b}) = q_3.$$

is equiloaded.

*Proof.* We shall prove the equiloadedness of the automaton $A$ using algebraic representation. The transition matrix $\Delta$ for the automaton $A$ is

$$\Delta = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

Now we prove that for $n > 0$ it holds that

$$\alpha_{2n} = \begin{pmatrix} 2^{n-1} \\ 0 \\ 2^{n-1} \\ 0 \end{pmatrix}, \quad \text{and} \quad \alpha_{2n+1} = \begin{pmatrix} 0 \\ 2^{n-1} \\ 0 \\ 2^{n-1} \end{pmatrix}. \tag{4.4}$$

We can easily see that the previous statement holds for $n = 1$. By induction we obtain

$$\alpha_{2n+2} = \Delta\alpha_{2n+1} = \begin{pmatrix} 2^n \\ 0 \\ 2^n \\ 0 \end{pmatrix}, \quad \alpha_{2n+3} = \Delta\alpha_{2n+2} = \begin{pmatrix} 0 \\ 2^n \\ 0 \\ 2^n \end{pmatrix}.$$

Next we shall look at $M_0, \ldots, M_4$ and we see that

$$M_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad M_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}, \quad M_2 = \begin{pmatrix} 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$M_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 3 & 3 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 3 & 1 & 1 & 3 \end{pmatrix}, \quad M_4 = \begin{pmatrix} 5 & 1 & 1 & 3 \\ 0 & 0 & 0 & 0 \\ 3 & 3 & 3 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Using (4.4) we can see that if $M_{2n}$ (as $M_4$ is) can be written (for some value $c_n$) as

$$M_{2n} = \begin{pmatrix} c_n + 2^{n-1} & c_n - 2^{n-1} & c_n - 2^{n-1} & c_n \\ 0 & 0 & 0 & 0 \\ c_n & c_n & c_n & c_n - 2^{n-1} \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

then the matrix $M_{2n+2}$ can be written as

$$M_{2n+2} = \begin{pmatrix} 2c_n + 2^n + 2^{n-1} & 2c_n - 2^{n-1} & 2c_n - 2^{n-1} & 2c_n + 2^{n-1} \\ 0 & 0 & 0 & 0 \\ 2c_n + 2^{n-1} & 2c_n + 2^{n-1} & 2c_n + 2^{n-1} & 2c_n - 2^{n-1} \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} c_{n+1} + 2^n & c_{n+1} - 2^n & c_{n+1} - 2^n & c_{n+1} \\ 0 & 0 & 0 & 0 \\ c_{n+1} & c_{n+1} & c_{n+1} & c_{n+1} - 2^n \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

(The matrices with odd indices are not interesting, because the number of words $w$ of length $2n+1$ such that $A$ ends in an accepting state while processing the word $w$ is zero)

We can see that $k = 3/4$ is good enough for definition of the equiloadedness. Indeed, for $n = 2l$ the maximally used state is $q_0$ with load $2c_l + 2^{l-1}$, the average load is $(8c_l - 2^l)/4$ and $|L \cap \Sigma^n| = 2^l$. □

## 4.8 Open Questions

### 4.8.1 Characterization

We left open the characterization of equiloaded automata. The algebraic representation of automata can be turned into computer program, which determines $k_i$, such that $A$ is equiloaded at words of the length $i$ with equiloadedness constant $k_i$. After examining matrices $\Delta$ with small size, we observed that if a sum of each row or column in $\Delta$ is equal to 2, and $\delta_{(i+1) \bmod n, i} \geq 1$, then $\{k_i\}_{i=0}^{\infty}$ seems to be convergent.

**Definition 4.3.** Let $\psi$ be a permutation of elements $0, 1, \ldots, n-1$. We shall say that a directed graph $G$ is *defined by the permutation* $\psi$, if

$$\forall i \in \{0, \ldots, n-1\} \quad (i, \psi(i)) \in E(G).$$

**Conjecture 4.1** (Two-cycles equiloaded automata)**.** Let $\psi_1, \ldots, \psi_k$ be permutations over $|Q|$ elements. Let $A$ be an automaton such that the graphical representation of $A$ is $G_0 \cup \bigcup_{i=0}^{k} G_i$, where $G_0 = (V, E_0)$ is an directed graph containing cycle through all states and $G_i = (V, E_i)$ is an directed graph defined by the permutation $\psi_i$. Then $A$ is equiloaded.

### 4.8.2 Equiloadedness Preserving Transformations

There are two automata transformations we conjecture to be equiloadedness preserving. First of them is a change of set of accepting states in such a way, that every state is used on some $w \in L(A)$.

The second transformation is $m$-regular splitting of automaton $A$ defined as follows.

**Definition 4.4.** Let $A = (Q, \Sigma, \delta_A, q_0, F)$ be an automaton, let $m$ be a nonzero natural number. By *an $m$-regular splitting* of automaton $A$ we obtain an automaton $B = (Q \times \{1, \dots, m\}, \Sigma, \delta_B, (q_0, 1), F_B)$, where $F_B \subseteq F \times \{1, \dots, m\}$ and for the transition function $\delta_B$ holds that

$$\forall p, q \in Q \ \exists i, j \in \mathbb{N} \setminus \{0\} \quad \delta_A(p, x) = q \Rightarrow \delta_B((p, i), x) = (q, j).$$

Moreover every state is accessible:

$$\forall p \in Q, \ \forall i \in \mathbb{N} \setminus \{0\} \quad \exists w \in L(B) \#[(p, i), w] > 0.$$

**Conjecture 4.2.** By an $m$-regular splitting of an equiloaded automaton $A$ we obtain an equiloaded automaton.

### 4.8.3 Sufficient Condition for Language to Be in $\mathcal{L}_{\mathrm{EQA}}$

In Theorem 4.3 we provide the necessary condition for automaton not to be equiloaded. We believe that converse implication holds as we state in the following conjecture. When the minimal-state automaton $A$ for a language fulfils the inequality (4.5) it suffices to split states of $A$ in such a way, that $A$ will be equiloaded. However, it is not clear that such a splitting is possible. In Example 4.3 we show an automaton for which such a splitting of states exists.

**Conjecture 4.3.** Let $L$ be a language. If for the minimal-state automaton $A$ for the language $L$ it holds that

$$\forall p, q \in Q \ \exists \ell \in \mathbb{R}^+ \ \exists n_0 \in \mathbb{N} \ \forall n \in \mathbb{N}, \ n > n_0 \quad \#[p, L \cap \Sigma^n] \leq \ell \#[q, L \cap \Sigma^n], \tag{4.5}$$

then $L \in \mathcal{L}_{\mathrm{EQA}}$.

*Example* 4.3. Consider the language $L = \{\mathrm{aaa}, \mathrm{bbb}\}^*$. The minimal-state automaton $A$ for this language is shown in Figure 4.6. This automaton is not equiloaded, as we showed in Example 4.1. However, it fulfils the inequality (4.5) because

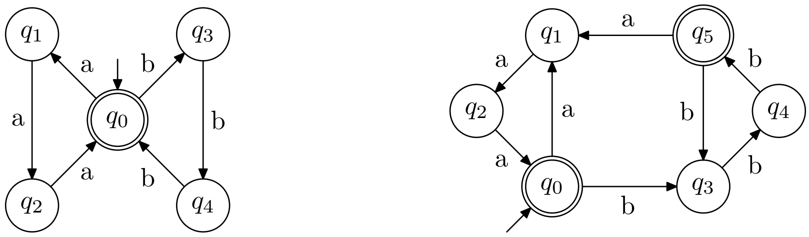
$$\frac{\#[q_0, L \cap \Sigma^{3k}]}{\#[q_1, L \cap \Sigma^{3k}]} = 2 + \frac{2}{k} \leq 3.$$

According to the above conjecture, there should be an equiloaded automaton $B$ that accepts the language $L$. Indeed, such an automaton exists as we can see in Figure 4.6. To show equiloadedness of the automaton $B$, we determine the load of the state $q_0$, which is used the most. On all words of length $n = 3m$, the load of $q_0$ is equal to triples consisting of a (a-triples) plus one (the initial use). The number of a-triples is between 0 and $m$. Furthermore, there are $m$ choose $i$ words of length $n$ with $i$ a-triples. Hence, the load of $q_0$ on words of length $3m$ is

$$\#[q_0, L \cap \Sigma^{3m}] = \sum_{i=0}^{m} \binom{m}{i}(i+1) = 2^{m-1}m + 2^m.$$

The automaton $B$ is equiloaded, because

$$2^{m-1}m + 2^m - \frac{(3m+1)2^m}{6} = \frac{5 \cdot 2^m}{6} = \frac{5}{6}|L \cap \Sigma^n|.$$




(a) A non-equiloaded automaton (b) An equiloaded automaton

Figure 4.6: (a) A non-equiloaded and (b) an equiloaded automaton for the language $L = \{\text{aaa}, \text{bbb}\}^*$

# Chapter 5

# Automata Equiloaded on Sequences of Words

In this chapter, we shall extend the previous definitions of equiloaded automata from words to sequences of words. According to Definition 3.1 the automaton is equiloaded, if it is equiloaded on each word. Sometimes, it is better to think about the load of an automaton while it is computing at many inputs. (Batch processing.) Definition 4.1 can be understood as an intermediate step to this point of view. It tells us that there can be some sequence of inputs, sorted by lengths of words, on which an automaton will load each state equally. As we will see in this chapter, Definition 4.1 can be taken as a special case of the definition of an automaton equiloaded on a sequence of words.

## 5.1 Definition

Firstly, we shall define a sequence of words. We shall only consider DFA that accept infinite languages. For our purposes, a sequence of words $S = w_1, w_2, \ldots$ is an infinite sequence of words from a language $L(A)$ accepted by a given automaton $A$. Moreover, words in a sequence will not be repeated, thus $w_i = w_j \Rightarrow i = j$.

We shall denote by $S(i, j)$ the subsequence of $S$ starting at the $i$-th word and ending at the $j$-th word.

**Definition 5.1.** Let $A = (Q, \Sigma, \delta, q_0, F)$ be a deterministic finite automaton. Let $S$ be a sequence of words from $L(A)$. The automaton $A$ is *equiloaded on S* if there exists an ascending function $f : \mathbb{N} \to \mathbb{N}$ and a function $k : \mathbb{N} \to \mathbb{R}^+$ such that for all $i \in \mathbb{N}$ and all $p, q \in Q$ it holds that

$$|\#[p, S(f(i), f(i+1)-1)] - \#[q, S(f(i), f(i+1)-1)]| \leq k(i)(f(i+1) - f(i)).$$

The function $k$ is said to be *an equiloadedness tolerance.*The function $f$ is said to be *a windows-defining function.*

## 5.2 Equivalent Form

*Notation.* The function $\operatorname{symb}(S(i,j)) = \sum_{w \in S(i,j)} |w|$ gives the number of symbols in the subsequence $S(i,j)$.

*Remark.* For the sake of brevity, we shall often use notation $f_i$ instead of $f(i)$ and $k_i$ instead of $k(i)$.

**Lemma 5.1.** A DFA $A = (Q, \Sigma, \delta, q_0, F)$ is equiloaded on a sequence $S$ if and only if there exists an ascending function $f : \mathbb{N} \to \mathbb{N}$ and a function $k' : \mathbb{N} \to \mathbb{R}^+$ such that for all $i \in \mathbb{N}$ it holds that

$$\max_{q \in Q} (\#[q, S(f_i, f_{i+1} - 1)]) - \operatorname{avg}(S(f_i, f_{i+1} - 1)) \leq k'_i(f_{i+1} - f_i), \tag{5.1}$$

where

$$\operatorname{avg}(S(i,j)) = \frac{\operatorname{symb}(S(i,j)) + j + 1 - i}{|Q|}$$

is the average load of states while $A$ processes words from the subsequence $S(i,j)$. Furthermore, there exists a $c \in \mathbb{R}^+$ such that $c \cdot k'(i) \geq k(i)$ for $k$ from Definition 5.1.

*Proof.* If $A$ is equiloaded on a sequence $S$, then similarly to previous definitions we consider the most- and the least-used state. It is easy to see that the average load of states on the subsequence $S(f_i, f_{i+1} - 1)$ is greater than or equal to the load of the least used state, hence the inequality (5.1) holds for $k_i = k'_i$.

Conversely, suppose that the inequality (5.1) holds. Then, for a given $i$, the load of each state is less than

$$\operatorname{avg}(S(f_i, f_{i+1} - 1)) + k'_i(f_{i+1} - f_i).$$

Also, the load of each state is greater than

$$\operatorname{avg}(S(f_i, f_{i+1} - 1)) - k'_i(f_{i+1} - f_i)(|Q| - 1).$$

Therefore, the difference between loads of arbitrary states $p$ and $q$ is not greater than

$$k'_i(f_{i+1} - f_i) + k'_i(|Q| - 1)(f_{i+1} - f_i) = k'_i \cdot |Q| \cdot (f_{i+1} - f_i).$$

Hence, automaton $A$ is equiloaded on the sequence $S$ with equiloadedness tolerance $k_i = k'_i|Q|$. □

**Theorem 5.1.** For a given automaton $A$, a sequence $S$, and an ascending function $f : \mathbb{N} \to \mathbb{N}$, there is a function $k : \mathbb{N} \to \mathbb{R}^+$ such that $A$ is equiloaded on the sequence $S$ with a windows-defining function $f$ and an equiloadedness tolerance $k$.

*Proof.* Define $k(n)$ by $k(i) = \max_{w \in S(f(i), f(i+1)-1)} |w| + 1$. We shall show, that $A$ is equiloaded on the sequence $S$ with the equiloadedness tolerance $k$. Let us bound from above the maximal load of a state at the subsequence $S(f(i), f(i+1)-1) = S_i$:

$$\#[q, S_i] \leq \mathrm{symb}(S_i) \leq (f(i+1) - f(i)) \cdot (\max_{w \in S_i} |w| + 1) = (f(i+1) - f(i)) \cdot k(i).$$

Hence, $A$ is equiloaded on the sequence $S$. □

Thus for a large equiloadedness tolerance the notion of equiloadedness becomes trivial. We shall therefore consider "reasonably small" $f$ and $k$ in what follows. It comes from the following theorem, that "reasonably small" in the case of the equiloadedness tolerance means constant, or at least sub-linear.*

**Theorem 5.2.** For every given automaton $A$ there exists a sequence $S$ of all words in $L(A)$ and a windows-defining function $f$ such that $A$ is equiloaded on $S$ with the equiloadedness tolerance $k(n) = \Omega(n)$.

*Proof.* The sequence we are looking for is the lexicographically ordered sequence $S$.† Let $f(i) = i$. This means, that we will consider windows of constant length 1. It is easy to see that

$$\#[q, S(i,i)] = \#[q, w_i] \leq |w_i| + 1 \leq |Q|i + 1.$$

The last inequality follows from the fact that if there is a word of length $n$ in an infinite language accepted by the automaton $A$, then in $A$ has to be a cycle of length at most $|Q|$, therefore there exists a word of length at most $n+|Q|$. From the above inequality we obtain that it suffices to take the equiloadedness tolerance $k(i) = |Q|i + 1 = \Omega(n)$. □

It should be possible to define the meaning of "reasonably small" value even for the case of the windows-defining function $f$. We conjecture that if an automaton is equiloaded on a sequence with a windows-defining function $f_1$ and a constant equiloadedness tolerance, then it is equiloaded on another sequence with a linear windows-defining function $f$ and a constant equiloadedness tolerance.

## 5.3 Impact of Order of Words on Equiloadedness Tolerance

We shall discuss the impact of choosing a sequence for an automaton $A$, on which we want $A$ to be equiloaded. As the first example, we shall consider an automaton $A = (\{q_0, q_1\}, \{\mathrm{a}, \mathrm{b}\}, \delta, q_0, \{q_0, q_1\})$, where the transition function $\delta$ is defined by

$$\delta(q_0, \mathrm{a}) = q_0, \quad \delta(q_0, \mathrm{b}) = q_1, \quad \delta(q_1, \mathrm{b}) = q_1.$$

Let $S$ be the lexicographically ordered sequence of words in the language $L(A)$, i.e., $S = \varepsilon, \mathrm{a}, \mathrm{b}, \mathrm{aa}, \mathrm{ab}, \mathrm{bb}, \ldots$

*Thus, in the following we shall consider only constant equiloadedness tolerance.

†It suffices to consider any sequence $S' = w_1, w_2, \ldots, w_n$ for which $|w_i| \leq |w_{i+1}|$.

**Theorem 5.3.** For the automaton $A$ and the sequence $S$ defined above, if $k(n)$ is a constant, and $A$ is equiloaded on $S$, then for any non-zero $l \in \mathbb{N}$ it holds $f(n) > ln$ for sufficiently large $n$, where $f$ is a windows-defining function from Definition 5.1. (The size of windows is greater than constant.)

*Proof.* For a given constant $k' = k(n)$ and $l$, we shall find a number $n$ such that $A$ is not equiloaded on $S(ln, ln + l - 1)$ with the equiloadedness tolerance $k'$. Let $n = 2lk' + 1$. Then the subsequence $S(ln, ln + l - 1) = S(2(lk')^2 + lk', 2(lk')^2 + lk' + l - 1)$ consists of words $\mathrm{a}^{2lk'}, \mathrm{a}^{2lk'-1}\mathrm{b}, \ldots \mathrm{a}^{2lk'-l+1}\mathrm{b}^{l-1}$. The load of the state $q_0$ on this subsequence is

$$\#[q_0, S(ln, ln+l-1)] = 2lk'+1+2lk'+2lk'-1+\cdots+2lk'-l+1 = (1/2)l(4lk'-l+3).$$

The load of $q_1$ is just $(1/2)l(l-1)$, therefore $A$ is not equiloaded with the equiloadedness tolerance $k'$. □

**Theorem 5.4.** Consider $A$ defined above. There is a sequence $S'$ of words in $L(A)$ such that $A$ is equiloaded on $S'$ with the equiloadedness tolerance $k(n) = 0$ and a linear windows-defining function $f$, $f(n) = 2n$.

*Proof.* If $f(n) = 2n$, the size of each window is equal to 2. We want to choose pairs of words from $L(A)$, such that the load of $q_0$ on the pair is equal to the load of $q_1$. First, we realize that by $\#[q_0, w]$ and $\#[q_1, w]$ $w$ is exactly determined: $w = \mathrm{a}^{\#[q_0,w]-1}\mathrm{b}^{\#[q_1,w]}$. We shall construct $S'$ in pairs as follows. For the $i$-th pair, we choose the first non-used word $w'_{2i}$ from $S$ and find a pair $w'_{2i+1}$, which is also a non-used word from $S$, such that $\#[q_0, w'_{2i}] + \#[q_0, w'_{2i+1}] = \#[q_1, w'_{2i}] + \#[q_1, w'_{2i+1}]$. The sequence $S'$ looks like $\varepsilon, \mathrm{bb}, \mathrm{a}, \mathrm{abb}, \mathrm{b}, \mathrm{abbb}, \ldots$ Then, $A$ is equiloaded on the sequence $S'$ with $k(n) = 0$ and $f(n) = 2n$. □

Now, we shall consider another automaton $B = (Q, \{\mathrm{a}, \mathrm{b}\}, \delta_B, q_0, q_0)$, a minimal automaton for a language $L = \{\mathrm{aaa}, \mathrm{bbb}\}^*$. This automaton, shown in Figure 4.6(a) in the previous chapter, is defined by transition function

$$\delta_B(q_0, \mathrm{a}) = q_1, \quad \delta_B(q_1, \mathrm{a}) = q_2, \quad \delta_B(q_2, \mathrm{a}) = q_0,$$
$$\delta_B(q_0, \mathrm{b}) = q_3, \quad \delta_B(q_3, \mathrm{b}) = q_4, \quad \delta_B(q_4, \mathrm{b}) = q_0.$$

It is easy to see that for all $w \in L$ it holds that

$$\#[q_0, w] = \#[q_1, w] + \#[q_2, w] + 1$$

We shall prove that there is no sequence $S$ such that $B$ is equiloaded on $S$ with constant equiloadedness tolerance (and arbitrary $f$).

**Theorem 5.5.** There is an automaton $B$ such that for any sequence $S$ and an ascending function $f$ it holds that $B$ is not equiloaded on $S$ with the windows-defining function $f$ and a constant equiloadedness tolerance.

*Proof.* Consider the automaton $B$ defined above and the subsequence of $S$ from $f(i)$ to $f(i+1)-1$. Without loss of generality, we may assume that $q_1$ is the least used state on that subsequence. Then it holds that

$$\begin{aligned} & \#[q_0, S(f_i, f_{i+1}-1)] - \#[q_1, S(f_i, f_{i+1}-1)] \geq \\ \geq\ & \frac{\mathrm{symb}(S(f_i, f_{i+1}-1))}{3} + (f_{i+1}-f_i) - \frac{\mathrm{symb}(S(f_i, f_{i+1}-1))}{6} \\ \geq\ & \frac{\mathrm{symb}(S(f_i, f_{i+1}-1))}{6} + (f_{i+1}-f_i) \\ =\ & (f_{i+1}-f_i)\left(\frac{\mathrm{symb}(S(f_i, f_{i+1}-1))}{6(f_{i+1}-f_i)}\right). \end{aligned}$$

Hence, the equiloadedness tolerance of $B$ on sequence $S$ should be at least

$$k(i) \geq \left(\frac{\mathrm{symb}(S(f_i, f_{i+1}-1))}{6(f_{i+1}-f_i)}\right) \notin \mathcal{O}(1).$$

It is easy to see that there is $i_0$ such that if $i > i_0$ the number of symbols in the subsequence is greater than the number of words in this subsequence. □

# Chapter 6

# Conclusion

In this thesis we discussed three approaches to the equiloadedness property for DFA. We have established a characterization of strictly equiloaded automata and, based on this characterization, we proved closure properties of the class $\mathcal{L}_{\mathrm{SEQA}}$.

We analyzed the characterization of equiloaded automata. After proving some basic results about equiloaded automata, we proved a necessary condition for a language to be in the family $\mathcal{L}_{\mathrm{EQA}}$. We introduced an algebraic representation which makes it easier to determine the equiloadedness of an automaton. We proved closure properties of the family $\mathcal{L}_{\mathrm{EQA}}$. Although we did not find a characterization based on a graphical representation of an automaton, we were able to formulate a conjecture on this characterization.

Next, we investigated equiloadedness property for sequences of words. We have shown that every automaton is equiloaded on every sequence given a sufficiently large equiloadedness tolerance. Furthermore, we proved that for each automaton there exists a sequence such that the automaton is equiloaded on the sequence with a linear equiloaded tolerance and a linear windows-defining function $f$. We have shown an existence of an automaton which is not equiloaded on any sequence with a constant equiloadedness tolerance.

Although we provide solutions to many interesting problems concerning equiloadedness, there are still some open problems, mainly in the area of equiloadedness on sequences of words. Among the most prominent are:

**Equiloaded automata:** a characterization of equiloaded automata and/or the family $\mathcal{L}_{\mathrm{EQA}}$, transformations preserving equiloadedness, a sufficient condition for a language to be in $\mathcal{L}_{\mathrm{EQA}}$.

**Automata equiloaded on sequences:** $f$-$k$ trade-off, operations over sequences.*

These questions may be an interesting topic for further research. Our results suggests that it may be worthwhile to study a balanced use of resources on a different model (NFA, push-down automata, ...) or with different resources (transitions between states).

*For this problem, we need to consider sequences of words from $L \subseteq L(A)$.

# Bibliography

[Hopcroft et al., 2001] Hopcroft, J. E., Motwani, R., and Ullman, J. D. (2001). *Introduction to automata theory, languages, and computation*. Addison-Wesley Longman Publishing, second edition.

[Hopcroft and Ullman, 1969] Hopcroft, J. E. and Ullman, J. D. (1969). *Formal languages and their relation to Automata*. Addison-Wesley Longman Publishing.

[Kováč, 2008] Kováč, I. (2008). O využívaní stavov v konečných automatoch. Bachelor's thesis, Faculty of Mathematics, Physics and Informatics, Comenius University, Bratislava.

[Nerode, 1958] Nerode, A. (1958). Linear automaton transformations. In *Proceedings of the American Mathematical Society*, 9.

[Rovan and Forišek, 2008] Rovan, B. and Forišek, M. (2008). Formálne jazyky a automaty. `http://foja.dcs.fmph.uniba.sk/materialy/skripta.pdf`.

# Abstrakt

| | |
|---|---|
| Autor: | Ivan Kováč |
| Názov práce: | Equiloaded Automata |
| Univerzita: | Univerzita Komenského v Bratislave |
| Fakulta: | Fakulta matematiky, fyziky a informatiky |
| Katedra: | Katedra informatiky |
| Školiteľ: | prof. RNDr. Branislav Rovan, PhD. |
| Počet strán: | 37 |
| Rok: | 2010 |

Predkladaná práca sa zaoberá štúdiom rovnomerného využívania prostriedkov vo výpočtoch. Uvažujeme konkrétny výpočtový model, deterministický konečný automat, a stavy takéhoto automatu ako prostriedok, ktorý sa má používať rovnomerne. V tomto modeli definujeme potrebné pojmy pre rovnomerné využívanie stavov a dokazujeme výsledky. V práci prezentujeme tri možné prístupy k rovnomernosti – *striktnú rovnomernosť*, *rovnomernosť* a *rovnomernosť na postupnostiach slov*. Analyzujeme triedy automatov a jazykov vzhľadom na tieto prístupy.

V práci prinášame charakterizáciu triedy jazykov, pre ktoré existuje striktne rovnomerný automat. Dokazujeme uzáverové vlastnosti tejto triedy.

Analyzujeme triedu jazykov, pre ktoré existuje rovnomerný automat, dokazujeme uzáverové vlastnosti tejto triedy, ako aj nutnú podmienku, ktorú musí jazyk spĺňať, aby do tejto triedy patril. Definujeme množinu transformácií automatov, ktoré zachovávajú rovnomernosť.

V súvislosti s rovnomernosťou na postupnostiach slov skúmame vplyv rôznych usporiadaní slov na mieru nerovnomernosti pre rovnomernosť na postupnostiach slov. Skúmame rovnomernosť na postupnostiach slov pre rôzne ohraničenia miery nerovnomernosti.

Naše výsledky môžu poslúžiť ako príklad pre podobný výskum pre iné výpočtové modely a prostriedky.

KĽÚČOVÉ SLOVÁ: rovnomerne využívané automaty, rovnomerné využívanie prostriedkov, deterministické konečné automaty