UPOZORNENIE

Tento súbor **nie je súčasťou diplomovej práce** ani jej oficiálnou prílohou. Ide o kópiu práce ŠVOČ 2013, ktorá je de facto rozšírenou verziou autorovej diplomovej práce. Dôvodom odoslania tejto práce spoločne s diplomovou prácou (iba prostredníctvom rozhrania na stránkach KI FMFI UK, do systému AIS tento súbor vložený nebol) je skutočnosť, že diplomová práca sa na túto (inak nezverejnenú) rozšírenú prácu často odvoláva. Autor práce sa preto domnieva, že by táto práca mala byť prístupná.

COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS

Balanced Use of Resources in Computations

ŠVOČ 2013

Author: Peter Kostolányi

Supervisor: prof. RNDr. Branislav Rovan, PhD.

University: Comenius University in Bratislava

Faculty: Faculty of Mathematics, Physics and Informatics

Abstract

Author:	Peter Kostolányi
Title:	Balanced Use of Resources in Computations
University:	Comenius University in Bratislava
Faculty:	Faculty of Mathematics, Physics and Informatics
Department:	Department of Computer Science
Supervisor:	prof. RNDr. Branislav Rovan, PhD.
Number of Pages:	165

Balanced use of resources in deterministic sequential computations is studied in this report. Several definitions of automata with balanced use of resources (equiloaded automata) are presented. These definitions are presented for abstract deterministic automata, i.e., an abstraction of automata devised for this purpose. Equiloadedness is then studied for three particular cases of abstract deterministic automata: deterministic finite automata, deterministic finite automata with ε-transitions, and deterministic one-counter automata. Several characterizations of families of equiloaded automata are proved. The families of languages accepted by equiloaded automata are also considered in the report.

KEYWORDS: balanced use of resources, equiloadedness, equiloaded automaton, abstract deterministic automaton.

Abstrakt

Autor:	Peter Kostolányi
Názov práce:	Rovnomerné využívanie prostriedkov vo výpočtoch (Balanced Use of Resources in Computations)
Univerzita:	Univerzita Komenského v Bratislave
Fakulta:	Fakulta matematiky, fyziky a informatiky
Katedra:	Katedra informatiky
Vedúci práce:	prof. RNDr. Branislav Rovan, PhD.
Rozsah:	165 strán

Práca sa zaoberá rovnomerným využívaním prostriedkov v deterministických sekvenčných výpočtoch. Predložených je niekoľko definícií automatov s rovnomerným využívaním prostriedkov (vyvážených automatov). Tieto definície sú formulované pre abstraktné deterministické automaty, t.j. pre abstrakciu automatov definovanú pre tento účel. Vyváženosť je potom predmetom štúdia pre tri špeciálne prípady abstraktných deterministických automatov: pre deterministické konečné automaty, deterministické konečné automaty s prechodmi na prázdne slovo a deterministické jednopočítadlové automaty. V práci je dokázaných viacero charakterizácií tried vyvážených automatov. Práca sa tiež zaoberá triedami jazykov akceptovaných vyváženými automatmi.

KĽÚČOVÉ SLOVÁ: rovnomerné využívanie prostriedkov, vyváženosť, vyvážený automat, abstraktný deterministický automat.

Contents

Abstract **iii**

Abstrakt **v**

Contents **viii**

Introduction **1**

1 Definitions and Basic Abstract Results **3**
- 1.1 Preliminaries 4
- 1.2 Abstract Deterministic Automata 6
- 1.3 Basic Quantities 10
- 1.4 Strict $\mathcal{S}$-Equiloadedness 10
- 1.5 $\mathcal{S}$-Equiloadedness 11
- 1.6 Relations between the Families of Equiloaded Languages 14
- 1.7 Prefix-Dense Languages and Strict $\mathcal{S}$-Equiloadedness 16

2 Deterministic Finite Automata **19**
- 2.1 Enumeration of Basic Quantities for DFA and DFAε 20
- 2.2 Strict $\mathcal{S}$-Equiloadedness 38
  - 2.2.1 Characterizations of Strict $\mathcal{S}$-Equiloadedness for $\mathcal{S} = \mathcal{C}$ and $\mathcal{S} = \mathcal{A}$ 39
  - 2.2.2 Relations between the Families of Strictly $\mathcal{S}$-Equiloaded Languages 42
  - 2.2.3 Closure Properties 44
- 2.3 $\mathcal{S}$-Equiloadedness 51
  - 2.3.1 Alternative Definitions of $\mathcal{S}$-Equiloadedness for $\mathcal{S} = \mathcal{C}_=$ and $\mathcal{S} = \mathcal{A}_=$ 51
  - 2.3.2 Computation of Basic Quantities via Convolutions 60
  - 2.3.3 Asymptotic Properties of Quantities for Strongly Connected Automata . . . 62
  - 2.3.4 Almost-Equivalence of Transition-Equiloadedness $\mathcal{S}$-Measures 68
  - 2.3.5 Characterizations of Weak $\mathcal{S}$-Equiloadedness for several $\mathcal{S}$ 73
  - 2.3.6 Relations between the Families of $\mathcal{S}$-Equiloaded Languages 88
  - 2.3.7 Closure Properties 94

3 Strictly $\mathcal{S}$-Equiloaded Deterministic One-Counter Automata **97**
- 3.1 Examples of Strictly $\mathcal{S}$-Equiloaded DOCA-Languages 97
- 3.2 Lemmas 100
- 3.3 Characterization of Strict Transition-$\mathcal{S}$-Equiloadedness 107
- 3.4 Further Lemmas 111
- 3.5 Families of Strictly $\mathcal{S}$-Equiloaded Languages 112
- 3.6 Closure Properties 120

Conclusion **127**

Bibliography **131**

A Mathematical Preliminaries **133**
A.1 Vandermonde and Generalized Vandermonde Matrices 133
A.2 Proving Linear Independence of Sequences 135
A.3 Systems of Linear OΔEs 137
A.4 Nonnegative Matrices 146
A.4.1 Irreducible Matrices 147
A.4.2 The Collatz-Wielandt Function 151
A.4.3 The Perron-Frobenius Eigenvalue 153
A.4.4 Dominating Nonnegative Matrix 158
A.4.5 The Perron-Frobenius Theorem 160
A.4.6 The Normal Form of a Reducible Matrix 161
A.5 Three Number-Theoretic Lemmas 163

Introduction

In this report, we shall study the concept of balanced use of resources in deterministic sequential computations. The problem of balancing the use of computational resources arises in several practical applications. However, up to now similar problems have received enough theoretical attention only in a parallel setting – the problem of balancing the load of computational tasks among several processors has been studied extensively. On the contrary, we are interested in sequential computations in this report, and our research may be viewed as a theoretical attempt to study computations with balanced use of multiple parts of a single processor.

Our approach to this problem is to study deterministic sequential models of computation and to come up with several possible definitions of automata with balanced use of resources. The resources considered are states and transitions. We shall call such automata *equiloaded*. We shall be interested in the properties of several families of equiloaded automata, and in the properties of families of languages accepted by these automata (*equiloaded languages*).

In this report, we continue the research initiated in several previous works. In [26] and [27], the balanced use of states in deterministic finite automata has been studied. In [25], we have studied the balanced use of transitions in deterministic finite automata.

The definitions of equiloadedness studied in this report are based on the definitions used in these previous works, however are significantly generalized. This generalization is twofold: first, the definitions presented in this report do not apply to deterministic finite automata only, but are defined for an abstract model of computation (*abstract deterministic automata*) that allows us to define equiloadedness for a variety of computational models at once. The second generalization has to do with sets of computations that we are concerned with while studying equiloadedness.

The main goals of this report therefore can be summarized as follows. First of all, we aim to present several sensible definitions of equiloadedness, suitable for diverse types of deterministic automata. Second, some of the definitions of state-equiloaded deterministic finite automata, used in [26] and [27], slightly differ from the definitions of transition-equiloaded deterministic finite automata, used in [25]. Thus, one of the aims of this report is to unify both theories by showing equivalence of certain definitions and by studying some aspects of equiloaded deterministic finite automata that have not been studied yet. Finally, since deterministic finite automata are the only model of computation, for which equiloadedness has been studied so far, we extend our study also to some other models of computation. In this report, we shall concentrate on equiloaded deterministic finite automata with ε-transitions (we shall observe that the possibility of ε-transitions adds some computational power to equiloaded deterministic finite automata) and initiate the study of equiloaded deterministic one-counter automata.

The structure of the report is as follows:

- *In Chapter 1*, after stating some preliminary definitions, we shall define an abstract model of computation inspired by abstract families of automata of S. Ginsburg [14] – an *abstract deterministic automaton*. We shall observe that some widely studied models of computation, e.g., deterministic finite automata, deterministic one-counter automata, or deterministic pushdown automata, are special cases of abstract deterministic automata.

  Further in this chapter, we shall present definitions of equiloadedness used in this report. These definitions will be stated for abstract deterministic automata, and thus will apply to a large variety of deterministic models of computation. Finally, we shall examine some properties of equiloadedness that hold for abstract deterministic automata in general.

- *In Chapter 2*, we shall focus on equiloaded deterministic finite automata. We shall unify the theories from [26], [27], and [25] and prove some new results. Moreover, we shall extend the theory to deterministic finite automata with ε-transitions.
- *In Chapter 3*, we shall initiate the study of equiloaded deterministic one-counter automata.
- *In Appendix A*, we shall briefly review some of the more advanced mathematics used in this report.

This report is in fact an extended version of a master's thesis that is to be submitted in June 2013. However, this report also contains some of the material that has been omitted from this thesis in order to preserve its reasonable length. Thus, the reader may find here also the proofs and explanatory material omitted from the thesis.

Chapter 1

Definitions and Basic Abstract Results

The main aim of this chapter is to present definitions of equiloadedness that we shall use in this report and to examine some of their basic properties that hold independently from a particular model of computation. We shall follow two conceptually different approaches to the definition of balacned use of resources. First, we shall define the concept of *strict $\mathcal{S}$-equiloadedness*, where $\mathcal{S}$ specifies the set of computation paths considered. Next, we shall define a slightly more involved concept of $\mathcal{S}$-equiloadedness and a related concept of weak $\mathcal{S}$-equiloadedness.

The chapter is structured as follows. In Section 1.1, we shall present some preliminary definitions that we shall use in this report. Most importantly, we shall present definitions of computational models that we shall study in this report: deterministic finite automata and deterministic one-counter automata. There is a need for including these definitions, since there are different variants of these models commonly used in literature. However, despite these differences usually do not matter, we shall observe that in the case of equiloaded automata, minor details in definitions of computational models may have significant consequences for their computational power.

In Section 1.2, we shall define an abstraction of deterministic automata with a one-way input tape (inspired by the concept of abstract families of automata [14]), the *abstract deterministic automata* (ADA). We shall observe that both deterministic finite automata and deterministic one-counter automata are special cases of ADA. The main reason for introducing this abstraction is that it enables us to define equiloadedness for a variety of computational models at once. That is, we shall present only one definition of each kind of equiloadedness that will apply to all models of computation considered in this report.

In Section 1.3, we shall briefly introduce some basic quantities that will serve as a cornerstone of our definitions of equiloadedness.

In Section 1.4, we shall define strictly $\mathcal{S}$-equiloaded automata. We shall state the definition independently from a particular computational model, i.e., for abstract deterministic automata. Here, $\mathcal{S}$ is a parameter specifying the set of computation paths considered.

In Section 1.5, we shall define $\mathcal{S}$-equiloaded automata. By slightly relaxing the conditions imposed in this definition, we shall obtain the related definition of weakly $\mathcal{S}$-equiloaded automata. Similarly as in the case of strict $\mathcal{S}$-equiloadedness, the definition is stated independently from a particular model of computation, and $\mathcal{S}$ specifies the sets of computation paths considered.

In Section 1.6, we shall examine some relations between the families of equiloaded languages that hold in general for every model of computation that is a special case of ADA.

Finally, in Section 1.7, we shall introduce a concept of prefix-dense languages. This concept will serve as a useful tool for proving that a given language is not strictly $\mathcal{S}$-equiloaded, for any computational model that is a special case of ADA.

1.1 Preliminaries

In this section, we shall briefly present some basic definitions used in this report. First, let us present our definition of deterministic finite automata. We shall be concerned with two different variants of deterministic finite automata: deterministic finite automata without ε-transitions (DFA) and deterministic finite automata with the possibility of deterministic ε-transitions (DFAε).

Definition 1.1.1 A *deterministic finite automaton with ε-transitions* (DFAε) A is a five-tuple $A = (K, \Sigma, \delta, q_0, F)$, where K is a nonempty finite set of states, Σ is an alphabet, $\delta : K \times (\Sigma \cup \{\varepsilon\}) \to K$ is a *partial* transition function that can be defined for (q, ε), q in K, only if $\delta(q, c)$ is not defined for any c in Σ, q_0 in K is the initial state, and $F \subseteq K$ is the set of accepting states. A *deterministic finite automaton without ε-transitions* (DFA) is a DFAε with a transition function not defined on ε.

Thus, there are two important facts that are necessary to keep in mind about our definition of deterministic finite automata. First, we are concerned with deterministic finite automata with a partial transition function, i.e., the transition function need not be defined for all possible inputs. Second, we are concerned with two different variants of deterministic finite automata, depending on if deterministic ε-transitions are allowed or not. Further, let us note that every DFA is at the same time a DFAε. Thus, if a theorem is stated for all DFAε, it holds also for all DFA. However, this is not true when talking about, e.g., the corresponding families of equiloaded languages.

The definitions of a configuration, a computation step and of the accepted language are standard. Moreover, we shall define a *transition* of a DFAε $A = (K, \Sigma, \delta, q_0, F)$ to be an arbitrary triple (p, c, q) in $K \times (\Sigma \cup \{\varepsilon\}) \times K$, such that $\delta(p, c) = q$. We shall denote the set of all transitions of the automaton A by D_A, or D.

Further, we shall define a *computation path* of the automaton A to be an arbitrary sequence of transitions corresponding to some computation of the automaton A. The formal definition is as follows.

Definition 1.1.2 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε. A *computation path* of the automaton A is a *finite* sequence $\gamma = \{(q_1, c_1, q_1'), \ldots, (q_n, c_n, q_n')\}$ of transitions in D, such that $q_1 = q_0$, and the property $q_{k+1} = q_k'$ holds for $k = 1, \ldots, n-1$. The number n is referred to as the *length* $|\gamma|$ *of the computation path* γ. A computation path γ is said to be *accepting*, if the state q_n' is accepting, i.e., if q_n' is in F.

We define the *graphical representation* of the automaton A to be a Σ-weighted digraph (with the possibility of multiple edges and loops) with the set of vertices corresponding to K, and with a c-weighted arc from a vertex p to a vertex q, if and only if (p, c, q) is in D. We shall define the *transition matrix* of the automaton A to be the adjacency matrix of its graphical representation. More formally, we shall define the transition matrix as follows.

Definition 1.1.3 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε. Let $K = \{q_0, \ldots, q_{m-1}\}$. The *transition matrix* of the automaton A is an $m \times m$ matrix

$$\Delta_A := \begin{pmatrix} d_{0,0} & d_{0,1} & \ldots & d_{0,m-1} \\ d_{1,0} & d_{1,1} & \ldots & d_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m-1,0} & d_{m-1,1} & \ldots & d_{m-1,m-1} \end{pmatrix},$$

where $d_{i,j}$ is defined by

$$d_{i,j} = \left| \{c \in \Sigma \cup \{\varepsilon\} \mid \delta(q_i, c) = q_j\} \right|,$$

for all i, j in $\{0, \ldots, m-1\}$. We shall omit the subscript A when A is clear from the context.

Now, let us define deterministic one-counter automata. We shall use a definition, in which a counter is represented by a single nonnegative integer (alternatively, a counter may be represented by a pushdown store over an unary alphabet).

Definition 1.1.4 A *deterministic one-counter automaton* (DOCA) A is a five-tuple $A = (K, \Sigma, \delta, q_0, F)$, where K is a nonempty finite set of states, Σ is an alphabet, $\delta : K \times (\Sigma \cup \{\varepsilon\}) \times \{0,1\} \to K \times \{-1,0,1\}$ is a deterministic (ε-transition may be defined only if there is not any other transition defined for given p in K and t in $\{0,1\}$) *partial* transition function, such that if $\delta(p, c, 0) = (q, r)$ for some p, q in K and c in $\Sigma \cup \{\varepsilon\}$, then r is in $\{0,1\}$, q_0 in K is the initial state, and $F \subseteq K$ is the set of accepting states.

To make the definition of deterministic one-counter automata absolutely clear, we shall also present formal definitions of a configuration, a computation step, and of the accepted language.

Definition 1.1.5 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DOCA. A *configuration* of the automaton A is a triple

$$(q, w, t) \text{ in } K \times \Sigma^* \times \mathbb{N},$$

where q is a state of the automaton A, w is an unread part of the input word, and t is a counter value of the automaton A.

Definition 1.1.6 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DOCA. A *computation step* of the automaton A is a relation $\vdash_A$ on configurations of A defined as follows:

$$(p, cw, t) \vdash_A (q, w, t') \iff \exists r \in \{-1, 0, 1\} : \delta(p, c, \operatorname{sgn}(t)) = (q, r) \wedge t' = t + r,$$

with p, q in K, c in $\Sigma \cup \{\varepsilon\}$, w in Σ^*, t, t' in $\mathbb{N}$, and with the function $\operatorname{sgn} : \mathbb{R} \to \mathbb{R}$ defined for all x in $\mathbb{R}$ by

$$\operatorname{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}.$$

If A is clear from the context, we shall write only $\vdash$ instead of $\vdash_A$. By $\vdash_A^*$ (resp. $\vdash^*$), we shall denote the reflexive and transitive closure of $\vdash_A$.

Definition 1.1.7 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DOCA. The *language* accepted by the automaton A is the set

$$L(A) = \{w \in \Sigma^* \mid \exists (q, t) \in F \times \mathbb{N} : (q_0, w, 0) \vdash_A^* (q, \varepsilon, t)\}.$$

Definition 1.1.8 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DOCA. The *language* accepted by the automaton A *by empty memory* is the set

$$N(A) = \{w \in \Sigma^* \mid \exists q \in K : (q_0, w, 0) \vdash_A^* (q, \varepsilon, \varepsilon)\}.$$

Finally, similarly as for deterministic finite automata, we shall define a transition and a computation path of a deterministic one-counter automaton.

Definition 1.1.9 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DOCA. A *transition* of the automaton A is a five-tuple

$$(q, c, t, q', r) \text{ in } K \times (\Sigma \cup \{\varepsilon\}) \times \{0,1\} \times K \times \{-1, 0, 1\},$$

such that $\delta(q, c, t) = (q', r)$. We shall denote the set of all transitions of the automaton A by D_A (resp. by D, if A is clear from the context).

Definition 1.1.10 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DOCA. A *computation path* of the automaton A is a *finite* sequence $\gamma = \{(q_1, c_1, t_1, q_1', r_1), \ldots, (q_n, c_n, t_n, q_n', r_n)\}$ of transitions of the automaton A, such that $q_1 = q_0$, $t_1 = 0$, and for $k = 1, \ldots, n-1$, the properties $q_{k+1} = q_k'$ and

$$t_{k+1} = \operatorname{sgn}\left(\sum_{i=1}^{k} r_i\right)$$

hold. The number n is referred to as the *length* $|\gamma|$ *of the computation path* γ. A computation path γ is said to be *accepting*, if the state q_n' is accepting, i.e., if q_n' is in F. A computation path γ is said to be *accepting by empty memory*, if $\sum_{i=1}^{n} r_i = 0$.

1.2 Abstract Deterministic Automata

In this section, we shall define a new abstract model of computation that is meant to serve, for the purposes of this report, as a simple generalization of several well-known and extensively studied deterministic computation models. We shall call these abstract automata *Abstract Deterministic Automata* (ADA).

Before we present a formal definition of abstract deterministic automata, let us briefly point out the importance of such a construction. In the next sections of this chapter, we shall define several types of equiloaded automata. For such definitions, some degree of independence from the computation model is a highly desired property. We would prefer to avoid separate definitions for each model of computation, and to devise, for each type of equiloadedness, one general definition applicable to a variety of interesting models of computation (e.g., deterministic finite automata, deterministic one-counter automata, deterministic pushdown automata, some variants of deterministic Turing machines, etc.). Since we shall be only concerned with models of computation that are special cases of ADA in this report, we shall consider definitions stated for ADA to be independent from a particular computational model.

Several other abstractions of automata are known up to date, for instance *Abstract Families of Automata* (AFA) [14], or *Balloon Automata* [20] [21]. However, both of these constructions are devised in order to guarantee certain properties of families of languages accepted by such automata and hence are, for the purposes of this report, unnecessarily complicated. Our definition of abstract deterministic automata does not guarantee these properties of accepted families of languages, but on the other hand, the definition is considerably simpler.

We shall define abstract deterministic automata as an abstraction of *deterministic one-way* automata. Therefore the requirements imposed on abstract deterministic automata are as follows:

- An ADA has a one-way input tape with ε-transitions allowed (that is, the input tape is read by the automaton exactly as in the case of, e.g., deterministic pushdown automata), and some kind of auxiliary memory storage.

- Although the auxiliary memory storage can acquire a possibly infinite number of memory contents, the transition function of the ADA distinguishes only a finite number of outputs of reading the auxiliary memory. The transitions of the ADA are executed based only on the information obtained by this output. For instance, in deterministic pushdown automata, the number of possible words on the pushdown store is infinite. However, the transition function distinguishes only between characters on the top of the pushdown store, i.e., members of a finite alphabet.

- An ADA is always deterministic – that is, for each state, input symbol, and output of reading the auxiliary memory, at most one transition can be executed.

This leads us to the formal definition that is as follows.

Definition 1.2.1 An *abstract deterministic automaton* (ADA) is a nine-tuple $A = (K, \Sigma, G, H, \zeta, Z, \delta, q_0, F)$, where K is a nonempty finite set of states, Σ is an input alphabet, G is a (finite or infinite) set of possible auxiliary memory contents, H is a finite set of outputs of reading the auxiliary memory, $\zeta : G \to H$ is a read function, Z in G is an initial content of the auxiliary memory, $\delta : K \times (\Sigma \cup \{\varepsilon\}) \times H \to K \times G^G$ is a *partial* transition function, q_0 in K is the initial state, and $F \subseteq K$ is the set of accepting states. Moreover, the transition function δ must be deterministic, i.e., the property

$$\forall p \in K \forall h \in H : (\exists (q, \mu) \in K \times G^G : \delta(p, \varepsilon, h) = (q, \mu)) \Rightarrow$$
$$\neg(\exists a \in \Sigma \exists (q', \mu') \in K \times G^G : \delta(p, a, h) = (q', \mu')).$$

is required to hold.

Let us explain the way the transition function δ is defined in the above definition. The transition function of an abstract deterministic automaton takes three arguments: a state, an input symbol (or ε, i.e., ε-transitions are allowed), and an output of reading the auxiliary memory. The transition function outputs a new state, and some transformation of the auxiliary memory μ. Every such transformation is a function from G to G, i.e., a member of the set G^G. We do not impose any restriction on these transformations, however we shall be interested only in cases, where these transformations are reasonable. The implication that is required to hold for the transition function δ ensures that only deterministic ε-transitions are allowed. Let us proceed by the definition of a configuration.

Definition 1.2.2 Let $A = (K, \Sigma, G, H, \zeta, Z, \delta, q_0, F)$ be an abstract deterministic automaton. A *configuration* of the automaton A is a triple (q, w, g) in $K \times \Sigma^* \times G$, where q is a state, w is an unread part of the input word, and g is a content of the auxiliary memory.

Now, let us define a computation step. This definition is more-or-less standard, given an intuitive idea of ADA presented above. An automaton is supposed to make a computation step from a given configuration (p, cw, g), where p is a state, c is a symbol (or the empty word ε) to be read from the input, and g is a content of the auxiliary memory. To make a computation step to a second configuration (q, w, g'), a transition function δ, given the state p, the symbol c, and the output $\zeta(g)$ after reading the content of the auxiliary memory g, has to return the state q and an auxiliary memory transformation μ, such that it transforms g into g', i.e., $\mu(g) = g'$. The definition of the computation step is thus as follows.

Definition 1.2.3 Let $A = (K, \Sigma, G, H, \zeta, Z, \delta, q_0, F)$ be an abstract deterministic automaton. A *computation step* of the automaton A is a binary relation $\vdash_A$ on configurations of the automaton A defined as follows:

$$(p, cw, g) \vdash_A (q, w, g') \iff \delta(p, c, \zeta(g)) = (q, \mu),$$

where p, q are in K, c is in $\Sigma \cup \{\varepsilon\}$, w is in Σ^*, g, g' are in G, and $\mu : G \to G$ is a mapping, such that $\mu(g) = g'$. If A is clear from the context, we shall write $\vdash$ instead of $\vdash_A$. By $\vdash_A^*$, we shall denote the reflexive and transitive closure of the relation $\vdash_A$.

Now, we may define the language accepted by a given abstract deterministic automaton. For several widely studied models of computation that are special cases of ADA, as for instance for deterministic one-counter automata or deterministic pushdown automata,[1] there is more then one mode of acceptation extensively studied: for instance, we may consider the language accepted by the accepting state, the language accepted by empty memory, etc. For ADA, we shall define two modes of acceptation: acceptation by the accepting state and acceptation by memory, with a possibility to model the acceptation by empty memory by the latter (at least for some special cases of ADA – for DFA for instance, acceptation by empty memory does not make much sense).

Definition 1.2.4 Let $A = (K, \Sigma, G, H, \zeta, Z, \delta, q_0, F)$ be an abstract deterministic automaton. The *language* accepted by the automaton A is the set

$$L(A) = \{w \in \Sigma^* \mid \exists (q, g) \in F \times G : (q_0, w, Z) \vdash_A^* (q, \varepsilon, g)\}.$$

Definition 1.2.5 Let $A = (K, \Sigma, G, H, \zeta, Z, \delta, q_0, F)$ be an abstract deterministic automaton, let $M \subseteq G$ be a set of memory contents. The *language* accepted by the automaton *A by memory* in M is a set

$$N_M(A) = \{w \in \Sigma^* \mid \exists (q, g) \in K \times M : (q_0, w, Z) \vdash_A^* (q, \varepsilon, g)\}.$$

[1]We consider the observation that these models of computation are indeed special cases of ADA, to be obvious. However, in the end of this section, we shall present some examples, in which we shall formally model some of the widely known models of computation in terms of ADA.

For some special cases of ADA, as for instance deterministic one-counter automata or deterministic pushdown automata, we define a special memory content 0_A, representing the empty memory. For such models of computation, we define the *language accepted by empty memory*, $N(A)$, to be the language accepted by memory in $\{0_A\}$. We shall call the special cases of ADA, for which we define the content 0_A, the models of computation *with the ability to accept by empty memory*. The definition of a transition is similar as, e.g., in the case of DFA.

Definition 1.2.6 Let $A = (K, \Sigma, G, H, \zeta, Z, \delta, q_0, F)$ be an abstract deterministic automaton. A *transition* of the automaton A is a five-tuple

$$(q, c, h, q', \mu) \in K \times (\Sigma \cup \{\varepsilon\}) \times H \times K \times G^G,$$

such that $\delta(q, c, h) = (q', \mu)$. We shall denote the set of all transitions of the automaton A by D_A (or by D, if A is clear from the context).

Finally, let us present the last definition related to abstract deterministic automata: the definition of a computation path. Similarly as in the case of DFA, a computation path is an arbitrary finite sequence of transitions corresponding to some computation of the automaton. This intuitive idea is formally expressed as follows.

Definition 1.2.7 Let $A = (K, \Sigma, G, H, \zeta, Z, \delta, q_0, F)$ be an abstract deterministic automaton. A *computation path* of the automaton A is a *finite* sequence

$$\gamma = \{(q_1, c_1, h_1, q_1', \mu_1), \ldots, (q_n, c_n, h_n, q_n', \mu_n)\} \in D_A^n$$

of n transitions of the automaton A, such that $q_1 = q_0$, and the following two properties hold:

(i) For $k = 1, \ldots, n-1$, the property $q_{k+1} = q_k'$ holds.

(ii) There is a sequence of auxiliary memory contents,

$$\mathbf{g} = \{g_1, \ldots, g_{n+1}\} \in G^{n+1},$$

such that:

1. $g_1 = Z$.
2. For $k = 1, \ldots, n$, the property $h_k = \zeta(g_k)$ holds.
3. For $k = 1, \ldots, n$, the property $\mu_k(g_k) = g_{k+1}$ holds.

The *length* $|\gamma|$ *of the computation path* γ is the number $|\gamma| = n$. A computation path γ is said to be *accepting*, if q_n' is in F. Moreover, if A has an ability to accept by empty memory, a computation path γ is said to be *accepting by empty memory*, if $g_{n+1} = 0_A$.

From the presented definitions, it should be clear that several extensively studied models of computation, including DFA, DFAε and DOCA, can be viewed as a special case of abstract deterministic automata. In what follows, we shall present formal constructions establishing this fact. For the sake of clarity of exposition, we shall avoid formal definitions of families of automata, and related concepts. Moreover, we shall use an intuitive notion of *isomorphism of automata*. We shall say that two automata are isomorphic, if all of their computations are isomorphic. However, we shall not formally define this notion of isomorphism, and shall rely on intuition. The notion of automata isomorphism is important, since automata defined in Section 1.1 are formally not the same tuples as abstract deterministic automata. However, we can clearly see if an abstract deterministic automaton behaves in its essence as one of the models of computation defined in Section 1.1.

Example 1.2.8 First, we shall show that deterministic finite automata and deterministic finite automata with ε-transitions are the special cases of abstract deterministic automata.

Let $A = (K, \Sigma, \delta, q_0, F)$ be a deterministic finite automaton with or without ε-transitions. Let us define the abstract deterministic automaton $A' = (K', \Sigma', G', H', \zeta', Z', \delta', q_0', F')$ isomorphic to A as follows: let us put down $K' = K$, $\Sigma' = \Sigma$, $G' = \{\circ\}$, $H' = \{\circ\}$. Next, the read function $\zeta' : G' \to H'$ is defined by $\zeta'(\circ) = \circ$, and $Z' = \circ$. The transition function $\delta' : K' \times (\Sigma' \cup \{\varepsilon\}) \times H' \to K' \times G'^{G'}$ is defined by

$$\delta'(p, c, \circ) = (q, Id) \iff \delta(p, c) = q,$$

where p, q are in $K = K'$, c is in $\Sigma \cup \{\varepsilon\} = \Sigma' \cup \{\varepsilon\}$, and $Id : G' \to G'$ is an identical mapping on G' (i.e., $Id(\circ) = \circ$). Finally, let us define $q_0' = q_0$, and $F' = F$.

Automata A and A' are clearly isomorphic. Moreover, A' has ε-transitions iff A has ε-transitions. That is, deterministic finite automata and deterministic finite automata with ε-transitions are both special cases of abstract deterministic automata.

Example 1.2.9 Now, we shall show that deterministic one-counter automata can be viewed as a special case of abstract deterministic automata. Let $A = (K, \Sigma, \delta, q_0, F)$ be a deterministic one-counter automaton. We shall construct the isomorphic abstract deterministic automaton $A' = (K', \Sigma', G', H', \zeta', Z', \delta', q_0', F')$ as follows: let us define $K' = K$, $\Sigma' = \Sigma$, $G' = \mathbb{N}$, $H' = \{0, 1\}$. The read function $\zeta' : G' \to H'$ will be defined by

$$\forall n \in \mathbb{N} : \zeta'(n) = \operatorname{sgn}(n).$$

Next, $Z' = 0$, and the transition function $\delta' : K' \times (\Sigma' \cup \{\varepsilon\}) \times H' \to K \times G'^{G'}$ will be defined by

$$\delta'(p, c, t) = (q, \mu_r) \iff \delta(p, c, t) = (q, r),$$

where p, q are in $K = K'$, c is in $\Sigma \cup \{\varepsilon\} = \Sigma' \cup \{\varepsilon\}$, t is in $\{0, 1\}$, r is in $\{-1, 0, 1\}$, and $\mu_r : G' \to G'$ is defined by

$$\forall n \in \mathbb{N} : \mu_r(n) = \max\{0, n + r\}.$$

Since the definition of deterministic one-counter automata ensures that if $t = 0$ then $r \geq 0$, the function μ_r always adds r to the counter (auxiliary memory). Finally, let us define $q_0' = q_0$, and $F' = F$.

Moreover, we shall define $0_{A'} = 0$. Thus, deterministic one-counter automata are able to accept by empty memory.

Thus, if we define concepts or state theorems for abstract deterministic automata in the rest of this report, we shall always keep in mind that the same concepts or theorems apply to all types of automata that we are concerned with in this report.

Finally in this section, let us introduce a notation for some special sets of computation paths that we shall use in this report.

Notation 1.2.10 Let A be an abstract deterministic automaton.

a) $\operatorname{Comp}(A)$ denotes the (possibly infinite) set of all computation paths of A.

b) $\operatorname{Comp}(A, n)$ denotes the (always finite) set of all computation paths of A of length n in $\mathbb{N}$.

c) $\operatorname{Comp}(A, \leq n)$ denotes the (always finite) set of all computation paths of A of length less than or equal to n in $\mathbb{N}$.

d) $\operatorname{Acc}(A)$ denotes the (possibly infinite) set of all accepting computation paths of A.

e) $\operatorname{Acc}(A, n)$ denotes the (always finite) set of all accepting computation paths of A of length n in $\mathbb{N}$.

f) $\mathrm{Acc}(A, \leq n)$ denotes the (always finite) set of all accepting computation paths of A of length less than or equal to n in $\mathbb{N}$.

Moreover, if A is an abstract deterministic automaton with an ability to accept by empty memory, we shall use the following notation:

g) $\mathrm{eAcc}(A)$ denotes the (possibly infinite) set of all computation paths of A accepting by empty memory.

h) $\mathrm{eAcc}(A, n)$ denotes the (always finite) set of all computation paths of A of length n in $\mathbb{N}$ accepting by empty memory.

i) $\mathrm{eAcc}(A, \leq n)$ denotes the (always finite) set of all computation paths of A of length less than or equal to n in $\mathbb{N}$ accepting by empty memory.

1.3 Basic Quantities

In this section, we shall briefly introduce some notation that will serve as a cornerstone of our definitions of equiloadedness, presented in the later sections of this chapter.

Notation 1.3.1 Let A be an ADA with the set of states K, and the set of transitions D. Let γ be a computation path of the automaton A, q in K be a state, and e in D be a transition. By the symbol $\#_A[q, \gamma]$, we shall denote the number of uses of the state q in the computation path γ. Similarly, by the symbol $\#_A[e, \gamma]$, we shall denote the number of uses of the transition e in the computation path γ.

Further, let Cmp be a *finite* set of computation paths of the automaton A. We shall use the notation

$$\#_A[q, Cmp] = \sum_{\gamma \in Cmp} \#_A[q, \gamma], \qquad \#_A[e, Cmp] = \sum_{\gamma \in Cmp} \#_A[e, \gamma].$$

If A is clear from the context, we shall omit the subscript A from the notation.

1.4 Strict $\mathcal{S}$-Equiloadedness

In this section, we shall present a definition of *strict $\mathcal{S}$-equiloadedness*, representing a first possible viewpoint on the nature of balanced use of resources. Informally, an automaton is strictly state-$\mathcal{S}$-equiloaded (strictly transition-$\mathcal{S}$-equiloaded), if its states (transitions) are used approximately the same number of times in every computation path from some set specified by $\mathcal{S}$. Thus, $\mathcal{S}$ is a parameter that specifies the set of computation paths considered.

Similar definitions have been used in [26], [27], and [25], in order to study balanced use of resources in DFA. However, the definition used in this report is more general. First, it is stated for ADA instead of DFA (this is also a reason for certain formal differences between the present definition and the definitions from the previous works). Second, the possibility to choose the parameter $\mathcal{S}$ is new – the definitions from the previous works have been concerned solely with the fixed set of all accepting computation paths.

The set $\mathcal{S}(A)$ of computation paths considered for a given automaton A is required to be specified for all ADA at once. Thus, $\mathcal{S}$ is a function defined on the family of all abstract deterministic automata that returns some set of computation paths of its input.

The formal definition is as follows.

Definition 1.4.1 Let $\mathcal{S}$ be a function that for each ADA A returns some set $\mathcal{S}(A)$ of computation paths of the automaton A. Let A be an ADA with the set of states K and the set of transitions D.

a) A is said to be *strictly state-$\mathcal{S}$-equiloaded*, if a real constant η in $\mathbb{R}$ exists, such that for all computation paths γ in $\mathcal{S}(A)$ and for all pairs of states p, q in K, the property

$$|\#[p,\gamma] - \#[q,\gamma]| \leq \eta$$

holds.

b) A is said to be *strictly transition-$\mathcal{S}$-equiloaded*, if a real constant η in $\mathbb{R}$ exists, such that for all computation paths γ in $\mathcal{S}(A)$ and for all pairs of transitions e, f in D, the property

$$|\#[e,\gamma] - \#[f,\gamma]| \leq \eta$$

holds.

Further, let x in $\{\mathrm{DFA}, \mathrm{DFA}\varepsilon, \mathrm{DOCA}, \ldots\}$ be an abbreviation of some model of computation that is a special case of ADA. A language L is said to be a *strictly state-$\mathcal{S}$-equiloaded x-language*, if a strictly state-$\mathcal{S}$-equiloaded ADA A of type x exists, such that $L(A) = L$. Similarly, a language L is said to be a *strictly transition-$\mathcal{S}$-equiloaded x-language*, if a strictly transition-$\mathcal{S}$-equiloaded ADA A of type x exists, such that $L(A) = L$.

We shall denote the family of strictly state-$\mathcal{S}$-equiloaded x-languages by $\mathscr{L}_{K-SEQ-x}(\mathcal{S})$. The family of all strictly transition-$\mathcal{S}$-equiloaded x-languages will be denoted by $\mathscr{L}_{\delta-SEQ-x}(\mathcal{S})$.

Similarly, a *strictly state-$\mathcal{S}$-equiloaded x-language accepted by empty memory*, and a *strictly transition-$\mathcal{S}$-equiloaded x-language accepted by empty memory* may be defined. We shall denote the family of strictly state-$\mathcal{S}$-equiloaded x-languages accepted by empty memory by $\mathscr{N}_{K-SEQ-x}(\mathcal{S})$, and the family of strictly transition-$\mathcal{S}$-equiloaded x-languages accepted by empty memory by $\mathscr{N}_{\delta-SEQ-x}(\mathcal{S})$.

In this report, we shall be concerned with three particular choices of $\mathcal{S}$ only. The first choice is $\mathcal{S} = \mathcal{C}$, defined for every ADA A by $\mathcal{C}(A) = \mathrm{Comp}(A)$, i.e., the set of *all* computation paths of the automaton A. The second choice is $\mathcal{S} = \mathcal{A}$, defined for every ADA A by $\mathcal{A}(A) = \mathrm{Acc}(A)$, i.e., the set of *all accepting* computation paths of the automaton A. Finally, the third choice is $\mathcal{S} = \mathcal{E}$, defined for every ADA A by $\mathcal{E}(A) = \mathrm{eAcc}(A)$, i.e., the set of *all* computation paths of the automaton A *accepting by empty memory*.

1.5 $\mathcal{S}$-Equiloadedness

In this section, we shall proceed to the definition of *$\mathcal{S}$-equiloadedness*, representing a formalization of a second, conceptually different idea of balanced use of resources. As in the case of strict $\mathcal{S}$-equiloadedness, the parameter $\mathcal{S}$ specifies a set of computation paths considered, however in a slightly different way.

In contrast with strict $\mathcal{S}$-equiloadedness, we shall not require the resources to be used approximately the same number of times in every computation path from $\mathcal{S}(A)$. Instead, we shall consider the (infinite) sequence of *finite* sets of computation paths

$$\{\mathcal{S}(A,n)\}_{n=0}^{\infty} = \{\mathcal{S}(A,0), \mathcal{S}(A,1), \mathcal{S}(A,2), \ldots\},$$

add up the numbers of uses of each resource in every of these finite sets, and require the resulting numbers of uses to be approximately the same in limit for $n \to \infty$. To make this intuitive requirement formal, we shall define the concept of *equiloadedness quotient* and *equiloadedness measure*.

The presented definition is a generalization of the definition that we have presented in [25] in order to study the balanced use of transitions in DFA. It also has a relation to definitions from [26] and [27]. This relation will become clear in Chapter 2, where we shall observe that these definitions are equivalent for DFA and DFAε.

In addition to $\mathcal{S}$-equiloadedness, we shall define also the concept of *weak $\mathcal{S}$-equiloadedness* (by a relaxation of the requirements imposed on $\mathcal{S}$-equiloaded automata).

The formal definitions of an equiloadedness quotient and of an equiloadedness measure are as follows.

Definition 1.5.1 Let $\mathcal{S}$ be a function which for each pair (A, n), A being an ADA and n a nonnegative integer, returns some finite set $\mathcal{S}(A, n)$ of computation paths of A. Let A be an ADA with the set of states K, and the set of transitions D. Let us denote $S_n := \mathcal{S}(A, n)$. Then we define the equiloadedness $\mathcal{S}$-quotients as follows:

a) The *n-th state-equiloadedness $\mathcal{S}$-quotient* of the automaton A is defined by:

$$\beta_A(\mathcal{S}, n) = \frac{\min_{p \in K} \#[p, S_n] + 1}{\max_{q \in K} \#[q, S_n] + 1}.$$

b) The *n-th transition-equiloadedness $\mathcal{S}$-quotient* of the automaton A is defined by:

$$B_A(\mathcal{S}, n) = \frac{\min_{e \in D} \#[e, S_n] + 1}{\max_{f \in D} \#[f, S_n] + 1}.$$

Moreover, we define the equiloadedness $\mathcal{S}$-measures as follows:

a) The (lower) *state-equiloadedness $\mathcal{S}$-measure* of an automaton A is defined by:

$$\beta_A(\mathcal{S}) = \liminf_{n \to \infty} \beta_A(\mathcal{S}, n).$$

b) The (lower) *transition-equiloadedness $\mathcal{S}$-measure* of an automaton A is defined by:

$$B_A(\mathcal{S}) = \liminf_{n \to \infty} B_A(\mathcal{S}, n).$$

The equiloadedness measure of an ADA is clearly a real number from the closed interval $[0, 1]$, with the value 1 representing the most balanced use of resources and the value 0 representing the least balanced use of resources. This motivates our definitions of *$\mathcal{S}$-equiloadedness* and *weak $\mathcal{S}$-equiloadedness*.

Definition 1.5.2 Let $\mathcal{S}$ be a function which for each pair (A, n), A being an ADA and n a nonnegative integer, returns some finite set $\mathcal{S}(A, n)$ of computation paths of A. Let A be an ADA.

a) A is said to be (weakly) state-$\mathcal{S}$-equiloaded, if $\beta_A(\mathcal{S}) = 1$ ($\beta_A(\mathcal{S}) > 0$).

b) A is said to be (weakly) transition-$\mathcal{S}$-equiloaded, if $B_A(\mathcal{S}) = 1$ ($B_A(\mathcal{S}) > 0$).

Further, let x in $\{\mathrm{DFA}, \mathrm{DFA}\varepsilon, \mathrm{DOCA}, \ldots\}$ be an abbreviation of some model of computation that is a special case of ADA. A language L is said to be a (weakly) *state-$\mathcal{S}$-equiloaded x-language*, if a (weakly) state-$\mathcal{S}$-equiloaded ADA A of type x exists, such that $L(A) = L$. Similarly, a language L is said to be a (weakly) *transition-$\mathcal{S}$-equiloaded x-language*, if a (weakly) transition-$\mathcal{S}$-equiloaded ADA A of type x exists, such that $L(A) = L$.

We shall denote the family of all (weakly) state-$\mathcal{S}$-equiloaded x-languages by the symbol $\mathscr{L}_{K-EQ-x}(\mathcal{S})$ ($\mathscr{L}_{K-WEQ-x}(\mathcal{S})$). Similarly, we shall denote the family of all (weakly) transition-$\mathcal{S}$-equiloaded x-languages by $\mathscr{L}_{\delta-EQ-x}(\mathcal{S})$ ($\mathscr{L}_{\delta-WEQ-x}(\mathcal{S})$).

Analogously, we define the families of (weakly) $\mathcal{S}$-equiloaded x-languages accepted by empty memory. The difference in the notation is in the use of the symbol $\mathscr{N}$ instead of $\mathscr{L}$.

In this report, we shall be concerned mainly by the following six particular choices of the parameter $\mathcal{S}$:

- $\mathcal{S}(A, n) = \mathcal{C}_=(A, n) = \mathrm{Comp}(A, n)$
- $\mathcal{S}(A, n) = \mathcal{E}_=(A, n) = \mathrm{eAcc}(A, n)$
- $\mathcal{S}(A, n) = \mathcal{A}_=(A, n) = \mathrm{Acc}(A, n)$
- $\mathcal{S}(A, n) = \mathcal{C}_\leq(A, n) = \mathrm{Comp}(A, \leq n)$

- $\mathcal{S}(A,n) = \mathcal{A}_{\leq}(A,n) = \mathrm{Acc}(A, \leq n)$
- $\mathcal{S}(A,n) = \mathcal{E}_{\leq}(A,n) = \mathrm{eAcc}(A, \leq n)$

In what follows, we shall state a lemma that provides us with an alternative formula for the computation of equiloadedness $\mathcal{S}$-measures. We shall use the lemma extensively in this report, since it makes the manipulation with equiloadedness $\mathcal{S}$-measures easier. For DFA and DFAε, the following lemma will allow us to numerically compute equiloadedness $\mathcal{S}$-measures.

Lemma 1.5.3 Let $\mathcal{S}$ be a function which for each pair (A,n), A being an ADA and n a nonnegative integer, returns some finite set $\mathcal{S}(A,n)$ of computation paths of A. Let A be an ADA with the set of states K, and the set of transitions D. Let us denote $S_n := \mathcal{S}(A,n)$. Then

$$\beta_A(\mathcal{S}) = \min_{(p,q)\in K^2} \liminf_{n\to\infty} \frac{\#[p,S_n]+1}{\#[q,S_n]+1}, \tag{1.1}$$

and

$$B_A(\mathcal{S}) = \min_{(e,f)\in D^2} \liminf_{n\to\infty} \frac{\#[e,S_n]+1}{\#[f,S_n]+1}. \tag{1.2}$$

Proof. We shall prove the equation (1.1) for states, the equation (1.2) for transitions can be proved in a similar manner. First, we shall prove that the left side of (1.1) is less than or equal to the right side of (1.1). Let p', q' in K be states of the automaton A. For all n in $\mathbb{N}$, we have

$$\#[p',S_n] \geq \min_{p\in K} \#[p,S_n],$$
$$\#[q',S_n] \leq \max_{q\in K} \#[q,S_n].$$

Thus, for all n in $\mathbb{N}$, the inequality

$$\beta_A(\mathcal{S},n) = \frac{\min_{p\in K}\#[p,S_n]+1}{\max_{q\in K}\#[q,S_n]+1} \leq \frac{\#[p',S_n]+1}{\#[q',S_n]+1}$$

holds. Thus,

$$\beta_A(\mathcal{S}) = \liminf_{n\to\infty} \beta_A(\mathcal{S},n) \leq \liminf_{n\to\infty} \frac{\#[p',S_n]+1}{\#[q',S_n]+1}. \tag{1.3}$$

Since (1.3) holds for all p', q' in K, we can clearly conclude

$$\beta_A(\mathcal{S}) \leq \min_{(p,q)\in K^2} \liminf_{n\to\infty} \frac{\#[p,S_n]+1}{\#[q,S_n]+1}.$$

Now, let us prove that the left side of (1.1) is greater than or equal to the right side of (1.1). Let us denote

$$m_n := \min_{p\in K} \#[p,S_n],$$
$$M_n := \max_{q\in K} \#[q,S_n].$$

For the purpose of contradiction, let us suppose that the inequality

$$\beta_A(\mathcal{S}) = \liminf_{n\to\infty} \frac{m_n+1}{M_n+1} < \min_{(p,q)\in K^2} \liminf_{n\to\infty} \frac{\#[p,S_n]+1}{\#[q,S_n]+1} =: \ell \tag{1.4}$$

holds. Let us denote

$$\lambda := \frac{\beta_A(\mathcal{S})+\ell}{2}.$$

From (1.4), it is possible to conclude that there is an infinite sequence $\{j_n\}_{n=0}^{\infty}$ of nonnegative integers, such that

$$\frac{m_{j_n}+1}{M_{j_n}+1} < \lambda$$

for all n in $\mathbb{N}$. Since the set of all pairs of states is finite, and since for all n in $\mathbb{N}$, a pair of states (p_n, q_n) in K^2 exists, such that

$$\#[p_n, S_n] = m_n, \qquad \text{and} \qquad \#[q_n, S_n] = M_n,$$

there is an infinite sequence $\{k_n\}_{n=0}^{\infty}$ of nonnegative integers, and states p', q' in K, such that

$$\frac{m_{j_{k_n}}+1}{M_{j_{k_n}}+1} = \frac{\#[p', S_{j_{k_n}}]+1}{\#[q', S_{j_{k_n}}]+1} < \lambda$$

holds for all n in $\mathbb{N}$. Thus, we have

$$\liminf_{n\to\infty} \frac{\#[p', S_n]+1}{\#[q', S_n]+1} \leq \lambda,$$

and, as a consequence,

$$\min_{(p,q)\in K^2} \liminf_{n\to\infty} \frac{\#[p, S_n]+1}{\#[q, S_n]+1} \leq \lambda,$$

which contradicts our assumption that

$$\min_{(p,q)\in K^2} \liminf_{n\to\infty} \frac{\#[p, S_n]+1}{\#[q, S_n]+1} = \ell.$$

The lemma is proved. □

1.6 Relations between the Families of Equiloaded Languages

In this section, we shall briefly examine some relations between the families of strictly $\mathcal{S}$-equiloaded and $\mathcal{S}$-equiloaded x-languages that hold for arbitrary x (being an abbreviation of some computational model that is a special case of ADA). These generally valid relations are rather basic. We shall prove more involved relations later in this report for particular models of computation.

Theorem 1.6.1 Let $\mathcal{S}$ be a function which for each pair (A, n), A being an ADA and n a nonnegative integer, returns some finite set $\mathcal{S}(A, n)$ of computation paths of the automaton A. Let x in $\{\text{DFA}, \text{DFA}\varepsilon, \text{DPDA}, \ldots\}$ be an abbreviation of some model of computation that is a special case of ADA. Then, the following inclusions hold:

1. $\mathscr{L}_{K-EQ-x}(\mathcal{S}) \subseteq \mathscr{L}_{K-WEQ-x}(\mathcal{S})$,
2. $\mathscr{L}_{\delta-EQ-x}(\mathcal{S}) \subseteq \mathscr{L}_{\delta-WEQ-x}(\mathcal{S})$,
3. $\mathscr{N}_{K-EQ-x}(\mathcal{S}) \subseteq \mathscr{N}_{K-WEQ-x}(\mathcal{S})$,
4. $\mathscr{N}_{\delta-EQ-x}(\mathcal{S}) \subseteq \mathscr{N}_{\delta-WEQ-x}(\mathcal{S})$.

(In the claims 3 and 4, it is assumed that automata of the type x have an ability to accept by empty memory.)

Proof. Let A be an ADA. If $\beta_A(\mathcal{S}) = 1$, then also $\beta_A(\mathcal{S}) > 0$. Thus, every state-$\mathcal{S}$-equiloaded ADA is also weakly state-$\mathcal{S}$-equiloaded. Similarly for transitions. □

Theorem 1.6.2 Let x in $\{\text{DFA}, \text{DFA}\varepsilon, \text{DPDA}, \ldots\}$ be an abbreviation of some model of computation that is a special case of ADA. Then, the following inclusions hold:

1. $\mathscr{L}_{K-SEQ-x}(\mathcal{C}) \subseteq \mathscr{L}_{K-EQ-x}(\mathcal{C}_=)$,
2. $\mathscr{L}_{\delta-SEQ-x}(\mathcal{C}) \subseteq \mathscr{L}_{\delta-EQ-x}(\mathcal{C}_=)$,
3. $\mathscr{L}_{K-SEQ-x}(\mathcal{A}) \subseteq \mathscr{L}_{K-EQ-x}(\mathcal{A}_=)$,
4. $\mathscr{L}_{\delta-SEQ-x}(\mathcal{A}) \subseteq \mathscr{L}_{\delta-EQ-x}(\mathcal{A}_=)$,
5. $\mathscr{N}_{K-SEQ-x}(\mathcal{C}) \subseteq \mathscr{N}_{K-EQ-x}(\mathcal{C}_=)$,
6. $\mathscr{N}_{\delta-SEQ-x}(\mathcal{C}) \subseteq \mathscr{N}_{\delta-EQ-x}(\mathcal{C}_=)$,
7. $\mathscr{N}_{K-SEQ-x}(\mathcal{E}) \subseteq \mathscr{N}_{K-EQ-x}(\mathcal{E}_=)$,
8. $\mathscr{N}_{\delta-SEQ-x}(\mathcal{E}) \subseteq \mathscr{N}_{\delta-EQ-x}(\mathcal{E}_=)$.

(It is assumed in the claims 5 – 8 that automata of the type x have an ability to accept by empty memory.)

Proof. We shall prove the theorem only for the case of transition-equiloadedness, the case of state-equiloadedness is analogous.

Let $\mathcal{S}_=$ be a function in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{E}_=\}$, and $\mathcal{S}$ be a function defined for every abstract deterministic automaton A by

$$\mathcal{S}(A) = \bigcup_{n=0}^{\infty} \mathcal{S}_=(A, n)$$

(i.e., if $\mathcal{S}_= = \mathcal{C}_=$, then $\mathcal{S} = \mathcal{C}$, and similarly for the other two choices of $\mathcal{S}_=$).

It clearly suffices to prove that if a given abstract deterministic automaton A with the set of transitions D is strictly transition-$\mathcal{S}$-equiloaded, then it is also transition-$\mathcal{S}_=$-equiloaded.

Since the number of transitions of any abstract deterministic automaton is finite, and since the identity

$$\sum_{e \in D} \#[e, \mathcal{S}_=(A, n)] = n \cdot |\mathcal{S}_=(A, n)|$$

holds, it follows from the Pigeonhole principle that

$$\max_{f \in D} \#[f, \mathcal{S}_=(A, n)] \geq \frac{n}{|D|} \cdot |\mathcal{S}_=(A, n)|. \tag{1.5}$$

Now, if the abstract deterministic automaton A is strictly transition-$\mathcal{S}$-equiloaded, the obvious inclusion $\mathcal{S}_=(A, n) \subseteq \mathcal{S}(A)$ implies

$$\begin{aligned}
\min_{e \in D} \#[e, \mathcal{S}_=(A, n)] &= \min_{e \in D} \sum_{\gamma \in \mathcal{S}_=(A,n)} \#[e, \gamma] \geq \max_{f \in D} \sum_{\gamma \in \mathcal{S}_=(A,n)} (\#[f, \gamma] - \eta) = \\
&= \left(\max_{f \in D} \sum_{\gamma \in \mathcal{S}_=(A,n)} \#[f, \gamma] \right) - \eta \cdot |\mathcal{S}_=(A, n)| = \\
&= \max_{f \in D} \#[f, \mathcal{S}_=(A, n)] - \eta \cdot |\mathcal{S}_=(A, n)|,
\end{aligned}$$

for some real constant η in $\mathbb{R}$. From this inequality, we obtain

$$\begin{aligned}
B_A(\mathcal{S}_=) &= \liminf_{n \to \infty} \frac{\min_{e \in D} \#[e, \mathcal{S}_=(A, n)] + 1}{\max_{f \in D} \#[f, \mathcal{S}_=(A, n)] + 1} \geq \\
&\geq \liminf_{n \to \infty} \frac{\max_{f \in D} \#[f, \mathcal{S}_=(A, n)] - \eta \cdot |\mathcal{S}_=(A, n)| + 1}{\max_{f \in D} \#[f, \mathcal{S}_=(A, n)] + 1} = \\
&= \liminf_{n \to \infty} \left(1 - \frac{\eta \cdot |\mathcal{S}_=(A, n)|}{\max_{f \in D} \#[f, \mathcal{S}_=(A, n)] + 1} \right) = 1,
\end{aligned}$$

if the automaton A is strictly transition-$\mathcal{S}$-equiloaded, since, by the inequality (1.5),

$$0 \leq \limsup_{n \to \infty} \frac{\eta \cdot |\mathcal{S}_=(A, n)|}{\max_{f \in D} \#[f, \mathcal{S}_=(A, n)] + 1} \leq \limsup_{n \to \infty} \frac{\eta \cdot |\mathcal{S}_=(A, n)|}{\frac{n}{|D|} \cdot |\mathcal{S}_=(A, n)| + 1} = 0,$$

i.e.,

$$\limsup_{n\to\infty} \frac{\eta \cdot |\mathcal{S}_=(A,n)|}{\max_{f\in D} \#[f,\mathcal{S}_=(A,n)]+1} = 0.$$

Thus, we have proved that if the automaton A is strictly transition-$\mathcal{S}$-equiloaded, then the inequality $B_A(\mathcal{S}_=) \geq 1$ holds. However, the converse inequality holds trivially. That is, $B_A(\mathcal{S}_=) = 1$, and the automaton A is transition-$\mathcal{S}_=$-equiloaded. The theorem is proved. □

Theorem 1.6.3 Let x in $\{\text{DFA}, \text{DFA}\varepsilon, \text{DPDA}, \ldots\}$ be an abbreviation of some model of computation that is a special case of ADA. Then, the following inclusions hold:

1. $\mathscr{L}_{K-SEQ-x}(\mathcal{C}) \subseteq \mathscr{L}_{K-SEQ-x}(\mathcal{A})$,
2. $\mathscr{L}_{\delta-SEQ-x}(\mathcal{C}) \subseteq \mathscr{L}_{\delta-SEQ-x}(\mathcal{A})$,
3. $\mathscr{N}_{K-SEQ-x}(\mathcal{C}) \subseteq \mathscr{N}_{K-SEQ-x}(\mathcal{E})$.
4. $\mathscr{N}_{\delta-SEQ-x}(\mathcal{C}) \subseteq \mathscr{N}_{\delta-SEQ-x}(\mathcal{E})$.

(It is assumed in the claims 3 – 4 that automata of the type x have an ability to accept by empty memory.)

Proof. It is obvious that for all abstract deterministic automata A of the type x, the inclusions $\mathcal{C}(A) \supseteq \mathcal{A}(A)$ and $\mathcal{C}(A) \supseteq \mathcal{E}(A)$ hold. Thus, if the inequality

$$|\#[p,\gamma] - \#[q,\gamma]| \leq \eta,$$

where η in $\mathbb{R}$ is a real constant holds for every two states p, q in K and every computation path γ in $\mathcal{C}(A)$, then it holds also for every computation path γ in $\mathcal{A}(A)$, and for every computation path γ in $\mathcal{E}(A)$.

Thus, if the automaton A is strictly state-$\mathcal{C}$-equiloaded, then it is also strictly state-$\mathcal{A}$-equiloaded and strictly state-$\mathcal{E}$-equiloaded. The same property can be analogously proved also for the case of transition-equiloadedness. The inclusions from the statement of the theorem then follow as a consequence. □

1.7 Prefix-Dense Languages and Strict $\mathcal{S}$-Equiloadedness

In this section, we shall introduce a notion of a prefix-dense language and connect this notion to the theory of strict $\mathcal{S}$-equiloadedness. The theory developed in this section will serve as a useful method for proving that certain languages are not strictly $\mathcal{S}$-equiloaded.

We shall achieve this proof method as follows: first, we shall prove that languages belonging to certain families of strictly $\mathcal{S}$-equiloaded languages are always prefix-dense (this will be done independently from the model of computation, i.e., for abstract deterministic automata). However, prefix-density of the language will be defined in such a manner that this property is usually easy to disprove for a given language.

Definition 1.7.1 Let L be an arbitrary language. The language L is said to be *prefix-dense*, if a nonnegative integer constant K in $\mathbb{N}$ exists, such that for every word w in L, the following property holds: let i, j in $\mathbb{N}$, $0 \leq i \leq j \leq |w|$, $j - i \geq K$, be nonnegative integers. Then, a nonnegative integer k in $\mathbb{N}$, $i \leq k \leq j$ exists, such that the prefix $w[1 \ldots k]$ of the word w is in L.

That is, the language L is said to be prefix-dense, if a constant K exists, such that for all words w from the language L, at least one of every $K+1$ consecutive prefixes of w is in L. In the following lemma, we shall prove an alternative description of prefix-density. This alternative description is useful, since it can be manipulated more easily.

Lemma 1.7.2 Let L be an arbitrary language. The language L is prefix-dense, if and only if a nonnegative integer constant K' in $\mathbb{N}$ exists, such that for all words w in L, the property

$$\mathrm{Pref}_{w,K'}(i) \cap L \neq \emptyset, \qquad i = 0, \ldots, \lfloor |w|/K' \rfloor \tag{1.6}$$

holds, where $\mathrm{Pref}_{w,K'}(i)$ is a language defined by

$$\mathrm{Pref}_{w,K'}(i) = \left\{ w[1\ldots k] \;\middle|\; k = iK', \ldots, (i+1)K' - 1 \right\}$$

for $i = 0, \ldots, \lfloor |w|/K' \rfloor - 1$, and by

$$\mathrm{Pref}_{w,K'}(i) = \left\{ w[1\ldots k] \;\middle|\; k = iK', \ldots, |w| \right\}$$

for $i = \lfloor |w|/K' \rfloor$.

Proof. If the language L is prefix-dense in the sense of Definition 1.7.1, for some constant K, then it clearly satisfies also the condition stated by this lemma, for the constant $K' = K + 1$: the language $\mathrm{Pref}_{w,K'}(\lfloor |w|/K' \rfloor)$ contains the word w, and thus (1.6) is satisfied for $i = \lfloor |w|/K' \rfloor$. The languages $\mathrm{Pref}_{w,K'}(i)$, $i = 0, \ldots, \lfloor |w|/K' \rfloor - 1$, consist each of $K' = K + 1$ consecutive prefixes of the word w. Thus, since Definition 1.7.1 is satisfied by assumption, the property (1.6) holds also for $i = 0, \ldots, \lfloor |w|/K' \rfloor - 1$.

Conversely, let the language L satisfy the condition imposed by this lemma, for some constant K'. It can be easily observed that it also satisfies the condition imposed by Definition 1.7.1, for $K = 2K'$. □

Before making a link to the theory of strict $\mathcal{S}$-equiloadedness, we shall state one more-or-less trivial observation:

Lemma 1.7.3 Let L be a finite language. Then it is prefix-dense.

Proof. Let l be the length of the longest word in the language L. Then, the condition imposed in Lemma 1.7.2 is clearly satisfied for $K' = l + 1$. □

Notation 1.7.4 We shall denote the family of all prefix-dense languages by $\mathscr{L}_{prefix}$.

The observation that is of crucial importance for the theory of strict $\mathcal{S}$-equiloadedness may be stated as follows:

Theorem 1.7.5 Let x in $\{\mathrm{DFA}, \mathrm{DFA}\varepsilon, \mathrm{DPDA}, \ldots\}$ be an abbreviation of some model of computation that is a special case of abstract deterministic automata. Let $\mathcal{S}$ be a function in $\{\mathcal{C}, \mathcal{A}\}$. Then, the following inclusions hold:

1. $\mathscr{L}_{K-SEQ-x}(\mathcal{S}) \subseteq \mathscr{L}_{prefix}$,
2. $\mathscr{L}_{\delta-SEQ-x}(\mathcal{S}) \subseteq \mathscr{L}_{prefix}$.

Proof. By Theorem 1.6.3, it suffices to prove the theorem for the case $\mathcal{S} = \mathcal{A}$. The remaining case $\mathcal{S} = \mathcal{C}$ is an immediate corollary.

Let us first prove that $\mathscr{L}_{K-SEQ-x}(\mathcal{A}) \subseteq \mathscr{L}_{prefix}$. For the purpose of contradiction, let us suppose that a language L in $\mathscr{L}_{K-SEQ-x}(\mathcal{A})$ exists, such that L is not prefix-dense (i.e., not in $\mathscr{L}_{prefix}$). Let $A = (K, \Sigma, G, H, \zeta, Z, \delta, q_0, F)$ be a strictly state-$\mathcal{A}$-equiloaded abstract deterministic automaton, such that $L(A) = L$.

Since the automaton A is strictly state-$\mathcal{A}$-equiloaded, a nonnegative real constant η in $\mathbb{R}$ exists, such that for all p, q in K and all accepting computation paths γ in $\mathrm{Acc}(A)$, the property

$$|\#[p, \gamma] - \#[q, \gamma]| \leq \eta$$

holds. Moreover, by Lemma 1.7.3, if the language L is finite, it is in $\mathscr{L}_{prefix}$. Thus, we may assume that the language L is infinite. Thus, the language L is also nonempty and therefore, at least one accepting state q_F in F exists.

Since the language L is not prefix-dense, it is clear that for every s in $\mathbb{N}$, words $u_s, v_s \in \Sigma^*$ exist, such that $u_s v_s$ is in L, $|v_s| \geq s$ and that there is no nonempty prefix w of v_s, such that $u_s w$ is in L.

Let γ_s be the accepting computation path of the automaton A on the word $u_s v_s$. Let γ' be the computation path of the automaton A on the word u_s (determinism implies that γ' is a prefix of γ). The property

$$\#[q_F, \gamma'] \leq \#[q, \gamma'] + \eta \tag{1.7}$$

has to hold for the computation path γ' and for all q in K. The reason for this is as follows: clearly, this property has to hold for the empty computation path. Moreover, it has to hold for every accepting prefix of the computation path γ' (of course, such a prefix need not exist). Let us denote by γ'' the longest such prefix (if there is not any, let us define γ'' to be the empty computation path). We have

$$\#[q_F, \gamma''] \leq \#[q, \gamma''] + \eta.$$

However, we also clearly have

$$\#[q_F, \gamma'] = \#[q_F, \gamma'']$$

and

$$\#[q, \gamma'] \geq \#[q, \gamma'']$$

for all q in K. Thus, the inequality (1.7) is proved. However, since the state q_F is accepting, we also have

$$\#[q_F, \gamma] = \#[q_F, \gamma']. \tag{1.8}$$

On the other hand, it follows from the Pigeonhole principle that for at least one p in K, we have

$$\#[p, \gamma] \geq \#[p, \gamma'] + \lfloor s/|K| \rfloor . \tag{1.9}$$

Thus, by (1.7), (1.8) and (1.9) we obtain

$$|\#[q_F, \gamma] - \#[p, \gamma]| \geq \lfloor s/|K| \rfloor - \eta.$$

However, since η is a constant and $\lfloor s/|K| \rfloor$ can be made arbitrarily high, this contradicts our assumption that the abstract deterministic automaton A is strictly state-$\mathcal{S}$-equiloaded.

The proof of the inclusion $\mathscr{L}_{\delta-SEQ-x}(\mathcal{A}) \subseteq \mathscr{L}_{prefix}$ is similar. The same train of thought can be followed, with the difference that instead in the number of uses of q_F, we would be interested in the number of uses of some transition leading to q_F. The details are left to the reader. $\square$

Example 1.7.6 The language $L_1 = \{a, b\}^*$ is clearly prefix-dense, since for every given w in L_1, every prefix of w is in L_1 as well.

However, we shall be more interested in languages that are not prefix-dense, since in that case we can also make negative statements about their strict $\mathcal{S}$-equiloadedness.

The language $L_2 = \{a^n b^n \mid n \geq 0\}$ is not prefix-dense, since for each n in $\mathbb{N}$, a counterexample $w_n = a^n b^n$ in L_2 exists, such that none of its $2n - 1$ proper prefixes

$$w_n[1 \ldots 1], w_n[1 \ldots 2], \ldots, w_n[1 \ldots 2n - 1]$$

is in L_2. For the similar reason, also the language $L_3 = \{a^n b \mid n \geq 1\} \cup \{\varepsilon\}$ is not prefix-dense. Thus, by Theorem 1.7.5, it follows that the languages L_2 and L_3 are not in any of the families $\mathscr{L}_{K-SEQ-x}(\mathcal{C})$, $\mathscr{L}_{K-SEQ-x}(\mathcal{A})$, $\mathscr{L}_{\delta-SEQ-x}(\mathcal{C})$, and $\mathscr{L}_{\delta-SEQ-x}(\mathcal{A})$, where x is an abbreviation of some computation model that is a special case of abstract deterministic automata.

However, if we are interested in acceptance by empty memory, prefix-density has only minor implications for strict $\mathcal{S}$-equiloadedness. In Chapter 3 (more precisely, in Examples 3.1.4 and 3.1.6), we shall show that the language L_2 is in $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$, and that L_3 is both in $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$ and in $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$.

Chapter 2

Deterministic Finite Automata

In the previous chapter, we have studied various aspects of several equiloadedness definitions for abstract deterministic automata. This general approach resulted in several useful observations that apply to all models of computation that can be viewed as a special case of ADA.

In this chapter, we begin the study of equiloadedness for particular models of computation and concentrate on *Deterministic Finite Automata* (with or without ε-transitions). A deterministic finite automaton is considered to be the simplest reasonable model of computation (at least among the models that are special cases of ADA) and is the only model of computation, for which the balanced use of resources has been studied up to now. In [26] and [27], some families of state-equiloaded DFA have been studied. In [25], we have initiated the study of transition-equiloaded DFA.

The definitions of equiloadedness used in [26], [27], and [25] slightly differ from those used in this report. However, using the terminology introduced in this report, we can say that in [26] and [27], families of strictly state-$\mathcal{A}$-equiloaded DFA and state-$\mathcal{A}_=$-equiloaded DFA have been studied[1] and in [25], we have studied the families of strictly transition-$\mathcal{A}$-equiloaded DFA, transition-$\mathcal{A}_=$-equiloaded DFA, and weakly transition-$\mathcal{A}_=$-equiloaded DFA.

In this chapter, we shall briefly restate the main results obtained in [26], [27], and [25] using the terminology and notation of this report (proofs will be omitted). However, the main focus will be on new results (those compose the great majority of this chapter). The definitions used in this report are far more general than the definitions used in [26], [27], and [25] and so it is desirable to give the known results an appropriate place in our theory. Therefore, this chapter will not consist of two separated parts, one consisting of the known results and one consisting of the new results. Instead of that, we shall be switching between the known and the new results. However, if a result is already known, a citation is always included. Therefore, it shall be relatively easy to distinguish between the known and the new results.

The new results presented later in this chapter can be roughly divided into the following categories:

- All known results have been about DFA, i.e., deterministic finite automata *without* ε-transitions. In this chapter, we shall study also DFAε, i.e., deterministic finite automata *with* ε-transitions.

- The definition of $\mathcal{S}$-equiloadedness presented in this report is far more general than related definitions used in [26], [27], and [25]. It can be proved that these older definitions are equivalent to $\mathcal{A}_=$-equiloadedness. However, in this chapter we shall study also other types of $\mathcal{S}$-equiloadedness. This involves also some new results on enumeration of basic quantities for DFA and DFAε, etc.

- A similar remark applies also to strict $\mathcal{S}$-equiloadedness.

[1] Up to now, the equivalence of the definition used in [26] and [27] and the definition of state-$\mathcal{A}_=$-equiloaded DFA has been an open problem. However, in this chapter we shall prove this equivalence.

- In this chapter, we shall solve an open problem concerning equivalence of certain definitions. This will lead to a unification of theories developed in [26] and [27], and in [25].
- We shall study weakly state-equiloaded DFA and DFAε that have not been studied up to now.

Before we concentrate on particular families of equiloaded DFA and DFAε, we shall show that several basic quantities used in the study of equiloaded finite automata may be computed relatively easily as solutions to certain initial value problems for systems of OΔEs (i.e., recurrences). Thus, we will be able to characterize the class of functions emerging in the numerator and the denominator of equiloadedness $\mathcal{S}$-quotients and thus to state several powerful results about equiloadedness $\mathcal{S}$-measures. Moreover, as a by-product, we will obtain a numerical algorithm for computing equiloadedness $\mathcal{S}$-measures for deterministic finite automata.

Afterwards, we shall first study strictly $\mathcal{S}$-equiloaded DFA and DFAε (Section 2.2), and subsequently $\mathcal{S}$-equiloaded (Section 2.3) DFA and DFAε for diverse choices of $\mathcal{S}$.

2.1 Enumeration of Basic Quantities for DFA and DFAε

In this section, we shall present systems of OΔEs (i.e., recurrences) that allow us to easily compute exact closed forms of several basic quantities used in the study of equiloaded DFA and DFAε, such as the number of computation paths of a given length, the number of uses of a given state or transition in computation paths of a given length, etc. These quantities form the basis of the theory of $\mathcal{S}$-equiloaded finite automata and the methods of their exact computation, presented in what follows, are crucial for the further developments of our theory.

To be more specific, we shall show that these basic quantities can be computed by solving certain initial value problems for homogeneous systems of first-order linear OΔEs with constant coefficients. Initial value problems of this kind can be solved relatively easily. For a general introduction to the topic, see, e.g., [10]. The method of solving systems of this kind is briefly reviewed in the end of this section. A brief treatment of the underlying theory, including the derivation of this method, can be found also in the appendix of this report.

The main idea behind the construction of systems presented in this subsection can be summarized as follows. For instance, one of the quantities we are interested in is the number of computation paths of length n. Let us slightly generalize this quantity, and let us consider the number of computation paths of length n beginning in a specified state q instead of q_0. More formally, let us denote by A_q the automaton identical to A except that its initial state is q. Then, this generalized quantity can be described as the number of computation paths of length n in the automaton A_q. Every such computation path is unambigously described by some transition leading from q and some computation path of length $n-1$ beginning in the resulting state of that transition. Thus, it is clear that the number of such computation paths can be computed as a sum through all transitions (q, c, q') in D of the numbers of computation paths of length $n-1$ in the automaton $A_{q'}$. These generalized quantities for all states of the automaton thus form a homogeneous system of first-order linear OΔEs with constant coefficients. The systems for other basic quantities can be derived in a similar manner.

As we shall observe, the matrices of the systems presented are always nonnegative, and thus can be transformed into the normal form of a reducible matrix (see, e.g., [31]). Later in this chapter, we shall make use of this fact and apply the Perron-Frobenius theory (see, e.g., [31]) to study the asymptotic properties of the basic quantities studied in this section.

Now we shall derive the systems for our basic quantities. While doing so, we shall use the following notation:

Notation 2.1.1 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε. Let q in K be a state. By A_q, we shall denote the deterministic finite automaton

$$A_q = (K, \Sigma, \delta, q, F),$$

i.e., the automaton A with the initial state replaced by q. Obviously, $A_{q_0} = A$.

Notation 2.1.2 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with $K = \{q_0, q_1, \ldots, q_{m-1}\}$. Let (q_j, c, q_k) in D be a transition, and q_l in K be a state. We shall use the following notation:

$$\begin{aligned}
F_i(n) &= |\mathrm{Comp}(A_{q_i}, n)|, && i = 0, 1, \ldots, m-1,\\
f_i(n) &= |\mathrm{Acc}(A_{q_i}, n)|, && i = 0, 1, \ldots, m-1,\\
G_i(n) &= |\mathrm{Comp}(A_{q_i}, \leq n)|, && i = 0, 1, \ldots, m-1,\\
g_i(n) &= |\mathrm{Acc}(A_{q_i}, \leq n)|, && i = 0, 1, \ldots, m-1,
\end{aligned}$$

$$\begin{aligned}
T_i^{(q_j,c,q_k)}(n) &= \#[(q_j, c, q_k), \mathrm{Comp}(A_{q_i}, n)], && i = 0, 1, \ldots, m-1,\\
t_i^{(q_j,c,q_k)}(n) &= \#[(q_j, c, q_k), \mathrm{Acc}(A_{q_i}, n)], && i = 0, 1, \ldots, m-1,\\
U_i^{(q_j,c,q_k)}(n) &= \#[(q_j, c, q_k), \mathrm{Comp}(A_{q_i}, \leq n)], && i = 0, 1, \ldots, m-1,\\
u_i^{(q_j,c,q_k)}(n) &= \#[(q_j, c, q_k), \mathrm{Acc}(A_{q_i}, \leq n)], && i = 0, 1, \ldots, m-1,
\end{aligned}$$

$$\begin{aligned}
S_i^{q_l}(n) &= \#[q_l, \mathrm{Comp}(A_{q_i}, n)], && i = 0, 1, \ldots, m-1,\\
s_i^{q_l}(n) &= \#[q_l, \mathrm{Acc}(A_{q_i}, n)], && i = 0, 1, \ldots, m-1,\\
V_i^{q_l}(n) &= \#[q_l, \mathrm{Comp}(A_{q_i}, \leq n)], && i = 0, 1, \ldots, m-1,\\
v_i^{q_l}(n) &= \#[q_l, \mathrm{Acc}(A_{q_i}, \leq n)], && i = 0, 1, \ldots, m-1.
\end{aligned}$$

That is, to compute the number of all (all accepting) computation paths of length n, it suffices to compute $F_0(n)$ ($f_0(n)$), and similarly for the rest of the quantities.

Now, we are finally prepared to state theorems about the systems for computing our basic quantities. We shall be interested in the eigenvalues of system matrices, since they are of key importance in the method of solving OΔEs of this type (see the end of this section, the appendix of this report, or the textbook [10]).

Theorem 2.1.3 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with $K = \{q_0, q_1, \ldots, q_{m-1}\}$. The functions

$$F_0, F_1, \ldots, F_{m-1}$$

are the unique solution to the initial value problem for the system of OΔEs

$$\mathbf{F}_n = \Delta \cdot \mathbf{F}_{n-1}, \qquad n \geq 1,$$

with $\mathbf{F}_n$ denoting a column vector

$$\mathbf{F}_n = (F_0(n), F_1(n), \ldots, F_{m-1}(n))^T$$

and with the initial conditions given by

$$\mathbf{F}_0 = (\underbrace{1, 1, \ldots, 1}_{m})^T.$$

Thus, the eigenvalues corresponding to this system are precisely the eigenvalues of the transition matrix Δ.

Proof. Let us first examine the trivial case $n = 0$. Clearly, for each k, there is exactly one computation path of length 0 beginning in q_k – in particular, the empty computation path. Thus, the initial conditions are indeed as in the statement of the theorem.

Now, let $n \geq 1$. Let k be fixed. We shall try to express $F_k(n)$ in terms of $F_i(n-1)$, with i in $\{0, \ldots, m-1\}$. Clearly, every computation path γ beginning in q_k must first go through some transition e beginning in q_k. This leads to some state q_i. Then, γ follows a computation path γ' of length $n-1$, beginning in q_i. Moreover, γ is clearly unambiguously determined by e and γ'.

Thus, $F_k(n)$ can be expressed as a sum of $F_i(n-1)$ for all i, such that there is a transition beginning in q_k, and ending in q_i (and each term in the sum is weighted by the number of such transitions). Formally,

$$F_k(n) = \sum_{(q_k,c,q_i)\in D} F_i(n-1) \tag{2.1}$$

(where q_k is fixed and the sum goes through all c and i). Now, if we write down the equation (2.1) for each k in $\{0,\ldots,m-1\}$, we obtain the system (with the sums going through all c and i)

$$\begin{aligned} F_0(n) &= \sum_{(q_0,c,q_i)\in D} F_i(n-1), \\ F_1(n) &= \sum_{(q_1,c,q_i)\in D} F_i(n-1), \\ &\vdots \\ F_{m-1}(n) &= \sum_{(q_{m-1},c,q_i)\in D} F_i(n-1). \end{aligned}$$

If we write this in the matrix-vector form, we obtain exactly the system

$$\mathbf{F}_n = \Delta \cdot \mathbf{F}_{n-1}, \qquad n \geq 1.$$

Thus, the theorem is proved. $\square$

A similar theorem may be stated also for the computation of functions enumerating the number of all accepting computation paths of a given length.

Theorem 2.1.4 Let $A = (K,\Sigma,\delta,q_0,F)$ be a DFAε with $K = \{q_0,q_1,\ldots,q_{m-1}\}$. The functions

$$f_0, f_1, \ldots, f_{m-1}$$

are the unique solution to the initial value problem for the system of OΔEs

$$\mathbf{f}_n = \Delta \cdot \mathbf{f}_{n-1}, \qquad n \geq 1,$$

with $\mathbf{f}_n$ denoting a column vector

$$(\mathbf{f}_n = f_0(n), f_1(n), \ldots, f_{m-1}(n))^T$$

and with the initial conditions given by

$$\mathbf{f}_0 = (C_0, C_1, \ldots, C_m)^T,$$

where

$$C_i = \begin{cases} 1 & \text{if } q_i \in F \\ 0 & \text{otherwise} \end{cases} \qquad i = 0,1,\ldots,m-1.$$

The eigenvalues corresponding to this system are, again, precisely the eigenvalues of the transition matrix Δ.

Proof. For each k, if q_k is accepting, then there is exactly one accepting computation path of length 0 beginning in q_k – the empty computation path. Otherwise, there is not any. Thus, the initial conditions are indeed as in the statement of the theorem. The rest may be proved in exactly the same way as in Theorem 2.1.3. $\square$

The following two systems may be used to enumerate the number of all (all accepting) computation paths of a length *at most n*.

Theorem 2.1.5 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with $K = \{q_0, q_1, \ldots, q_{m-1}\}$. The functions

$$G_0, G_1, \ldots, G_{m-1}, F_0, F_1, \ldots, F_{m-1}$$

are the unique solution to the initial value problem for the system of OΔEs

$$\mathbf{G}_n = \begin{pmatrix} \mathbf{I}_m & \Delta \\ \mathbf{0} & \Delta \end{pmatrix} \cdot \mathbf{G}_{n-1}, \qquad n \geq 1,$$

where $\mathbf{I}_m$ denotes the $m \times m$ identity matrix, $\mathbf{0}$ denotes the $m \times m$ zero matrix, $\mathbf{G}_n$ denotes a column vector with $2m$ entries

$$\mathbf{G}_n = (G_0(n), G_1(n), \ldots, G_{m-1}(n), F_0(n), F_1(n), \ldots, F_{m-1}(n))^T,$$

and where the initial conditions are given by

$$\mathbf{G}_0 = (\underbrace{1, 1, \ldots, 1, 1, 1, \ldots, 1}_{2m})^T.$$

Since the system matrix is an upper triangular block matrix, from its form it is obvious that if Δ has eigenvalues $\lambda_1, \ldots, \lambda_k$ with multiplicities $\alpha_1, \ldots, \alpha_k$, then the block matrix of the system has exactly these eigenvalues plus m times the eigenvalue one.

Proof. The number of all computation paths of length at most n can be clearly expressed as a number of all computation paths of length exactly n plus the number of all compuation paths of length at most $n-1$. That is, we have

$$\begin{aligned} G_0(n) &= G_0(n-1) + F_0(n) \\ G_1(n) &= G_1(n-1) + F_1(n) \\ &\vdots \\ G_{m-1}(n) &= G_{m-1}(n-1) + F_{m-1}(n) \end{aligned}$$

with initial conditions

$$G_i(0) = 1, \qquad i = 0, 1, \ldots, m-1.$$

This can be viewed either as a nonhomogeneous system with m unknown functions, or after expressing the functions $F_0, \ldots, F_{m-1}$ from the system presented in Theorem 2.1.3, as a homogeneous system with $2m$ unknown functions

$$G_0, G_1, \ldots, G_{m-1}, F_0, F_1, \ldots, F_{m-1}.$$

In the latter case, the system can be written in a (block) matrix form as

$$\mathbf{G}_n = \begin{pmatrix} \mathbf{I}_m & \Delta \\ \mathbf{0} & \Delta \end{pmatrix} \cdot \mathbf{G}_{n-1}, \qquad n \geq 1,$$

with the initial conditions given by

$$\mathbf{G}_0 = (\underbrace{1, 1, \ldots, 1, 1, 1, \ldots, 1}_{2m})^T.$$

Thus, the theorem is proved. □

Theorem 2.1.6 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with $K = \{q_0, q_1, \ldots, q_{m-1}\}$. The functions

$$g_0, g_1, \ldots, g_{m-1}, f_0, f_1, \ldots, f_{m-1}$$

are the unique solution to the initial value problem for the system of OΔEs

$$\mathbf{g}_n = \begin{pmatrix} \mathbf{I}_m & \Delta \\ \mathbf{0} & \Delta \end{pmatrix} \cdot \mathbf{g}_{n-1}, \qquad n \geq 1,$$

where $\mathbf{I}_m$ denotes the $m \times m$ identity matrix, $\mathbf{0}$ denotes the $m \times m$ zero matrix, $\mathbf{g}_n$ denotes a column vector with $2m$ entries

$$\mathbf{g}_n = (g_0(n), g_1(n), \ldots, g_{m-1}(n), f_0(n), f_1(n), \ldots, f_{m-1}(n))^T.$$

The initial conditions are given by

$$\mathbf{g}_0 = (C_0, C_1, \ldots, C_{m-1}, D_0, D_1, \ldots, D_{m-1})^T,$$

where

$$C_i = D_i = \begin{cases} 1 & \text{if } q_i \in F \\ 0 & \text{otherwise} \end{cases} \qquad i = 0, 1, \ldots, m-1.$$

Since the system matrix is an upper triangular block matrix, from its form it is obvious that if Δ has eigenvalues $\lambda_1, \ldots, \lambda_k$ with multiplicities $\alpha_1, \ldots, \alpha_k$, then the block matrix of the system has exactly these eigenvalues plus m times the eigenvalue one.

Proof. The proof is analogous as in the case of Theorem 2.1.5. □

In what follows, we shall present the systems for the computation of the number of uses of a given transition in certain computation paths.

Theorem 2.1.7 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with $K = \{q_0, q_1, \ldots, q_{m-1}\}$. Let (q_j, c, q_k) in D be a transition. The functions

$$T_0^{(q_j,c,q_k)}, T_1^{(q_j,c,q_k)}, \ldots, T_{m-1}^{(q_j,c,q_k)}, F_0, F_1, \ldots, F_{m-1}$$

are the unique solution to the initial value problem for the system of OΔEs

$$\mathbf{T}_n = \begin{pmatrix} \Delta & * \\ \mathbf{0} & \Delta \end{pmatrix} \cdot \mathbf{T}_{n-1}, \qquad n \geq 1,$$

where $\mathbf{0}$ is the $m \times m$ zero matrix, $*$ is the placeholder for an arbitrary $m \times m$ matrix,[2] $\mathbf{T}_n$ denotes a column vector with $2m$ entries

$$\mathbf{T}_n = \left(T_0^{(q_j,c,q_k)}(n), T_1^{(q_j,c,q_k)}(n), \ldots, T_{m-1}^{(q_j,c,q_k)}(n), F_0(n), F_1(n), \ldots, F_{m-1}(n) \right)^T,$$

and the initial conditions are given by

$$\mathbf{T}_0 = (\underbrace{0, 0, \ldots, 0}_{m}, \underbrace{1, 1, \ldots, 1}_{m})^T.$$

It is clear that the block matrix of the presented system has exactly the same eigenvalues as the matrix Δ, although with doubled multiplicities.

Proof. If $n = 0$, $T_i^{(q_j,c,q_k)}(0)$ is clearly 0 for $i = 0, \ldots, m-1$, since no transition is used in computation paths of length 0.

Let $n \geq 1$. Every computation path γ of length n beginning in some state q_i can be decomposed into a transition, and a computation path γ' of length $n-1$ beginning in a state determined by that transition. If $i \neq j$, then the transition (q_j, c, q_k) is not among the transitions leading from

[2] Although we could be able to exactly characterize this block, it is irrelevant for the purposes of this report.

q_i, and the number of uses of the transition (q_j, c, q_k) can be therefore computed as a sum of the numbers of uses of that transition in all possible computation paths γ'. If $i = j$, then the total number of uses of the transition (q_j, c, q_k) consists of this sum plus the number of computation paths γ beginning with the transition (q_j, c, q_k).

This leads us to the system (with the sums going through all c and i)

$$\begin{aligned}
T_0^{(q_j,c,q_k)}(n) &= \sum_{(q_0,c,q_i)\in D} T_i^{(q_j,c,q_k)}(n-1) \\
T_1^{(q_j,c,q_k)}(n) &= \sum_{(q_1,c,q_i)\in D} T_i^{(q_j,c,q_k)}(n-1) \\
&\vdots \\
T_{j-1}^{(q_j,c,q_k)}(n) &= \sum_{(q_{j-1},c,q_i)\in D} T_i^{(q_j,c,q_k)}(n-1) \\
T_j^{(q_j,c,q_k)}(n) &= F_k(n-1) + \sum_{(q_j,c,q_i)\in D} T_i^{(q_j,c,q_k)}(n-1) \\
T_{j+1}^{(q_j,c,q_k)}(n) &= \sum_{(q_{j+1},c,q_i)\in D} T_i^{(q_j,c,q_k)}(n-1) \\
&\vdots \\
T_{m-1}^{(q_j,c,q_k)}(n) &= \sum_{(q_{m-1},c,q_i)\in D} T_i^{(q_j,c,q_k)}(n-1)
\end{aligned}$$

with initial conditions

$$T_i^{(q_j,c,q_k)}(0) = 0 \qquad \forall i \in \{0, 1, \ldots, m-1\}.$$

This can be viewed either as a nonhomogeneous system with m unknown functions, or as a homogeneous system with $2m$ unknown functions

$$T_0^{(q_j,c,q_k)}, T_1^{(q_j,c,q_k)}, \ldots, T_{m-1}^{(q_j,c,q_k)}, F_0, F_1, \ldots, F_{m-1},$$

incorporating the system from Theorem 2.1.3. In the latter case, the system can be written in a (block) matrix form as in the statement of the theorem. Thus, the theorem is proved. □

Theorem 2.1.8 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with $K = \{q_0, q_1, \ldots, q_{m-1}\}$. Let (q_j, c, q_k) in D be a transition. The functions

$$t_0^{(q_j,c,q_k)}, t_1^{(q_j,c,q_k)}, \ldots, t_{m-1}^{(q_j,c,q_k)}, f_0, f_1, \ldots, f_{m-1}$$

are the unique solution to the initial value problem for the system of OΔEs

$$\mathbf{t}_n = \begin{pmatrix} \Delta & * \\ \mathbf{0} & \Delta \end{pmatrix} \cdot \mathbf{t}_{n-1}, \qquad n \geq 1,$$

where $\mathbf{0}$ is the $m \times m$ zero matrix, $*$ is the placeholder for an arbitrary $m \times m$ matrix, $\mathbf{t}_n$ denotes a column vector with $2m$ entries

$$\mathbf{t}_n = \left(t_0^{(q_j,c,q_k)}(n), t_1^{(q_j,c,q_k)}(n), \ldots, t_{m-1}^{(q_j,c,q_k)}(n), f_0(n), f_1(n), \ldots, f_{m-1}(n) \right)^T,$$

and the initial conditions are given by

$$\mathbf{t}_0 = (\underbrace{0, 0, \ldots, 0}_{m}, C_0, C_1, \ldots, C_{m-1})^T,$$

where

$$C_i = \begin{cases} 1 & \text{if } q_i \in F \\ 0 & \text{otherwise} \end{cases} \qquad i = 0, 1, \ldots, m-1.$$

It is clear that the block matrix of the presented system has exactly the same eigenvalues as the matrix Δ, although with doubled multiplicities.

Proof. The proof is analogous to the proof of Theorem 2.1.7. □

Theorem 2.1.9 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with $K = \{q_0, q_1, \ldots, q_{m-1}\}$. Let (q_j, c, q_k) in D be a transition. The functions

$$U_0^{(q_j,c,q_k)}, U_1^{(q_j,c,q_k)}, \ldots, U_{m-1}^{(q_j,c,q_k)}, T_0^{(q_j,c,q_k)}, T_1^{(q_j,c,q_k)}, \ldots, T_{m-1}^{(q_j,c,q_k)}, F_0, F_1, \ldots, F_{m-1}$$

are the unique solution to the initial value problem for the system of OΔEs

$$\mathbf{U}_n = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix} \cdot \mathbf{U}_{n-1}, \qquad n \geq 1,$$

where $\mathbf{I}_m$ is the $m \times m$ identity matrix, $\mathbf{0}$ is the $m \times m$ zero matrix, $*$ is the placeholder for an arbitrary $m \times m$ matrix, and where $\mathbf{U}_n$ denotes a column vector with $3m$ entries

$$\mathbf{U}_n = \Big(U_0^{(q_j,c,q_k)}(n), U_1^{(q_j,c,q_k)}(n), \ldots, U_{m-1}^{(q_j,c,q_k)}(n), T_0^{(q_j,c,q_k)}(n), T_1^{(q_j,c,q_k)}(n), \ldots, T_{m-1}^{(q_j,c,q_k)}(n), F_0(n), F_1(n), \ldots, F_{m-1}(n) \Big)^T.$$

The initial conditions are given by

$$\mathbf{U}_0 = (\underbrace{0, 0, \ldots, 0}_{m}, \underbrace{0, 0, \ldots, 0}_{m}, \underbrace{1, 1, \ldots, 1}_{m})^T.$$

Clearly, if the transition matrix Δ has eigenvalues $\lambda_1, \ldots, \lambda_k$ with respective multiplicities $\alpha_1, \ldots, \alpha_k$, then the eigenvalues of the system matrix are[3]

$$2\alpha_1 \times \lambda_1, \ldots, 2\alpha_k \times \lambda_k + m \times 1.$$

Proof. We consider the statement to be clear. □

Theorem 2.1.10 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with $K = \{q_0, q_1, \ldots, q_{m-1}\}$. Let (q_j, c, q_k) in D be a transition. The functions

$$u_0^{(q_j,c,q_k)}, u_1^{(q_j,c,q_k)}, \ldots, u_{m-1}^{(q_j,c,q_k)}, t_0^{(q_j,c,q_k)}, t_1^{(q_j,c,q_k)}, \ldots, t_{m-1}^{(q_j,c,q_k)}, f_0, f_1, \ldots, f_{m-1}$$

are the unique solution to the initial value problem for the system of OΔEs

$$\mathbf{u}_n = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix} \cdot \mathbf{u}_{n-1}, \qquad n \geq 1,$$

[3] In our symbolic notation (that is also used later in Table 2.1), $\alpha \times \lambda$ means that the eigenvalue λ has the algebraic multiplicity α. The operator $+$ can be viewed as a union on a multiset representing the spectrum, i.e., if S is a spectrum, $S + \alpha \times \lambda$ is S with the multiplicity of the eigenvalue λ increased by α (the original multiplicity of λ in S could have been zero or nonzero).

where $\mathbf{I}_m$ is the $m \times m$ identity matrix, $\mathbf{0}$ is the $m \times m$ zero matrix, $*$ is the placeholder for an arbitrary $m \times m$ matrix, and where $\mathbf{u}_n$ denotes a column vector with $3m$ entries

$$\mathbf{u}_n = \Big(u_0^{(q_j,c,q_k)}(n), u_1^{(q_j,c,q_k)}(n), \ldots, u_{m-1}^{(q_j,c,q_k)}(n), t_0^{(q_j,c,q_k)}(n), t_1^{(q_j,c,q_k)}(n), \ldots, t_{m-1}^{(q_j,c,q_k)}(n),$$
$$f_0(n), f_1(n), \ldots, f_{m-1}(n) \Big)^T.$$

The initial conditions are given by

$$\mathbf{u}_0 = (\underbrace{0,0,\ldots,0}_{m}, \underbrace{0,0,\ldots,0}_{m}, D_0, D_1, \ldots, D_{m-1})^T,$$

where

$$D_i = \begin{cases} 1 & \text{if } q_i \in F \\ 0 & \text{otherwise} \end{cases} \qquad i = 0, 1, \ldots, m-1.$$

Since the matrix of the system is the same as in the previous case, the eigenvalues are the same as well. That is, if the transition matrix Δ has eigenvalues $\lambda_1, \ldots, \lambda_k$ with respective multiplicities $\alpha_1, \ldots, \alpha_k$, then the eigenvalues of the system matrix are

$$2\alpha_1 \times \lambda_1, \ldots, 2\alpha_k \times \lambda_k + m \times 1.$$

Proof. The statement is clear. □

The matrices and corresponding eigenvalues for the above studied systems are summarized in Table 2.1. For the rest of the basic quantities, we shall not explicitly construct systems of OΔEs (although it is certainly possible), but we shall express these quantities in terms of quantities, for which we already have systems constructed.

Next, we shall present a theorem on the enumeration of the number of uses of a given *state* in all (or all accepting) computation paths of length n ($\leq n$). The functions for these quantities shall be expressed in terms of functions counting the number of uses of a given transition.

Theorem 2.1.11 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with $K = \{q_0, q_1, \ldots, q_{m-1}\}$. Let q_l in K be a state. Then the following equations hold for $q_l \neq q_0$:

$$S_0^{q_l}(n) = \sum_{(q_k,c,q_l) \in D} T_0^{(q_k,c,q_l)}(n),$$
$$s_0^{q_l}(n) = \sum_{(q_k,c,q_l) \in D} t_0^{(q_k,c,q_l)}(n),$$
$$V_0^{q_l}(n) = \sum_{(q_k,c,q_l) \in D} U_0^{(q_k,c,q_l)}(n),$$
$$v_0^{q_l}(n) = \sum_{(q_k,c,q_l) \in D} u_0^{(q_k,c,q_l)}(n).$$

The following equations hold for $q_l = q_0$:

$$S_0^{q_0}(n) = F_0(n) + \sum_{(q_k,c,q_0) \in D} T_0^{(q_k,c,q_0)}(n),$$
$$s_0^{q_0}(n) = f_0(n) + \sum_{(q_k,c,q_0) \in D} t_0^{(q_k,c,q_0)}(n),$$
$$V_0^{q_0}(n) = G_0(n) + \sum_{(q_k,c,q_0) \in D} U_0^{(q_k,c,q_0)}(n),$$

Quantity	Function	Matrix	Eigenvalues
$\lvert\mathrm{Comp}(A,n)\rvert$	$F_0(n)$	Δ	$\alpha_1 \times \lambda_1, \ldots, \alpha_k \times \lambda_k$
$\lvert\mathrm{Acc}(A,n)\rvert$	$f_0(n)$	Δ	$\alpha_1 \times \lambda_1, \ldots, \alpha_k \times \lambda_k$
$\lvert\mathrm{Comp}(A,\leq n)\rvert$	$G_0(n)$	$\begin{pmatrix} \mathbf{I}_m & \Delta \\ \mathbf{0} & \Delta \end{pmatrix}$	$\alpha_1 \times \lambda_1, \ldots, \alpha_k \times \lambda_k + m \times 1$
$\lvert\mathrm{Acc}(A,\leq n)\rvert$	$g_0(n)$	$\begin{pmatrix} \mathbf{I}_m & \Delta \\ \mathbf{0} & \Delta \end{pmatrix}$	$\alpha_1 \times \lambda_1, \ldots, \alpha_k \times \lambda_k + m \times 1$
$\#[(q_j,c,q_k),\mathrm{Comp}(A,n)]$	$T_0^{(q_j,c,q_k)}(n)$	$\begin{pmatrix} \Delta & * \\ \mathbf{0} & \Delta \end{pmatrix}$	$2\alpha_1 \times \lambda_1, \ldots, 2\alpha_k \times \lambda_k$
$\#[(q_j,c,q_k),\mathrm{Acc}(A,n)]$	$t_0^{(q_j,c,q_k)}(n)$	$\begin{pmatrix} \Delta & * \\ \mathbf{0} & \Delta \end{pmatrix}$	$2\alpha_1 \times \lambda_1, \ldots, 2\alpha_k \times \lambda_k$
$\#[(q_j,c,q_k),\mathrm{Comp}(A,\leq n)]$	$U_0^{(q_j,c,q_k)}(n)$	$\begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix}$	$2\alpha_1 \times \lambda_1, \ldots, 2\alpha_k \times \lambda_k + m \times 1$
$\#[(q_j,c,q_k),\mathrm{Acc}(A,\leq n)]$	$u_0^{(q_j,c,q_k)}(n)$	$\begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix}$	$2\alpha_1 \times \lambda_1, \ldots, 2\alpha_k \times \lambda_k + m \times 1$

Table 2.1: Summary of matrices and corresponding eigenvalues for systems of OΔEs presented in this section. Sets of eigenvalues and multiplicities for the specified matrices are listed in terms of eigenvalues of the transition matrix Δ, denoted $\lambda_1, \ldots, \lambda_k$, and their multiplicities, $\alpha_1, \ldots, \alpha_k$. By m, we denote the number of states of the automaton A, i.e., the transition matrix Δ is of the type $m \times m$.

$$v_0^{q_0}(n) = g_0(n) + \sum_{(q_k,c,q_0) \in D} u_0^{(q_k,c,q_0)}(n),$$

(the sums go through all q_k in K and c in $\Sigma \cup \{\varepsilon\}$).

Proof. We consider the theorem to be obvious. □

Of course, an analogous theorem holds also for $S_i^{q_l}, s_i^{q_l}, V_i^{q_l}$, and $v_i^{q_l}$, for $i = 1, \ldots, m-1$. However, we are not interested in these functions.

It is a direct corollary of the presented theorem and of the method for solving initial value problems of our kind that solutions for functions counting the number of uses of a given state have the same form (the solutions differ only in constant coefficients) as the solutions for functions counting the number of uses of a given transition. Thus, it essentially does not matter if we are enumerating the number of uses of a given state or a given transition – the solution has always the same form, and the only difference is in the values of the constant coefficients occurring in this solution.

Now, let us briefly outline the method of solving initial value problems for homogeneous systems of first-order linear OΔEs with constant coefficients (more details can be found, e.g., in [10], or in the appendix of this report). A system of this kind can be always written in a matrix-vector form

$$\mathbf{x}_n = M \cdot \mathbf{x}_{n-1}, \qquad n \geq 1, \tag{2.2}$$

where M is an $m \times m$ matrix, and for every n in $\mathbb{N}$, $\mathbf{x}_n$ is a column vector with m entries

$$\mathbf{x}_n = (x_0(n), \ldots, x_{m-1}(n))^T.$$

In an initial value problem, the vector $\mathbf{x}_0$ is given explicitly as a vector of initial conditions

$$\mathbf{x}_0 = (C_0, \ldots, C_{m-1})^T.$$

In this report, we are not interested in solving initial value problems precisely as described above, since instead of the closed form for $\mathbf{x}_n$, we are interested in the closed form of $x_0(n)$ only. This allows us to avoid the computation of eigenvectors, or possibly generalized eigenvectors (without loosing anything). The method of finding the closed form of $x_0(n)$ can be summarized as follows:

1. Write down the system in the matrix-vector form (2.2).

2. Compute eigenvalues of the matrix M. Let us denote distinct nonzero eigenvalues of the matrix M by $\lambda_1, \ldots, \lambda_k$, and their algebraic multiplicities by $\alpha_1, \ldots, \alpha_k$. Let us denote by α the multiplicity of a zero eigenvalue.

3. The solution for $x_0(n)$ has a form
$$x_0(n) = \sum_{i=1}^{k} \sum_{j=0}^{\alpha_i - 1} c_{i,j} \cdot n^j \lambda_i^n + \sum_{j=0}^{\alpha - 1} c_{n=j} \cdot [n = j] \tag{2.3}$$
for some constants $c_{i,j}, i = 1, \ldots, k, j = 0, \ldots, \alpha_i - 1$ and $c_{n=j}, j = 0, \ldots, \alpha - 1$ to be determined.

4. Determine the unknown constants in (2.3) by solving the system of linear equations obtained from the initial conditions.

Thus, we may conclude that we have successfully developed a method for enumeration of all basic quantities listed in the beginning of this subsection. Moreover, by Lemma 1.5.3, the computation of equiloadedness $\mathcal{S}$-measures is (at least for several most important choices of $\mathcal{S}$) reduced to the computation of a lower limit of a certain very special form. That is, the methods presented in this subsection may be turned into a numerical algorithm for computing equiloadedness $\mathcal{S}$-measures.[4]

Example 2.1.12 Now, we shall demonstrate the established method on an example. Let us consider a deterministic finite automaton $A = (K, \Sigma, \delta, q_0, F)$ with $K = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $F = \{q_2\}$, and with the transition function δ defined by

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

It is clear that the transition matrix Δ of the deterministic finite automaton A is

$$\Delta = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix},$$

and that the automaton A accepts the language $L = \{aa\}^*\{b\}\{a, b\}^*$.

We shall compute some of the basic quantities (namely the functions counting the number of computation paths, the number of uses of the transition (q_1, a, q_0), and the number of uses of the state q_0) for this automaton.

[4]Probably the least trivial step is the computation of eigenvalues. However, eigenvalues can be numerically computed by using, e.g., the QR algorithm (for details, see, e.g., [15]).

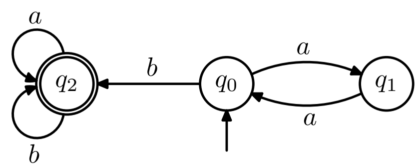


Figure 2.1: The automaton A accepting the language $L = \{aa\}^*\{b\}\{a,b\}^*$. The basic quantities are being computed for this automaton.

Let us first compute the eigenvalues of the transition matrix Δ. By computing the characteristic polynomial of this matrix, we obtain

$$\mathrm{ch}(\lambda) = \begin{vmatrix} -\lambda & 1 & 1 \\ 1 & -\lambda & 0 \\ 0 & 0 & 2-\lambda \end{vmatrix} = (2-\lambda)(\lambda-1)(\lambda+1).$$

Thus, the eigenvalues of Δ are

$$\lambda_1 = 2,$$
$$\lambda_2 = 1,$$
$$\lambda_3 = -1,$$

and (since all of these eigenvalues are simple) the corresponding algebraic multiplicities are

$$\alpha_1 = 1,$$
$$\alpha_2 = 1,$$
$$\alpha_3 = 1.$$

Since there is no zero eigenvalue, the multiplicity α of the zero eigenvalue is $\alpha = 0$, thus we are allowed to omit the second sum in the general solution.

The general solutions for the basic quantities follow directly from the results established earlier in this section. It is therefore clear, that the general solutions for these quantities can be expressed in the form

$$\begin{aligned}
F_0(n) &= c^{(1)}_{1,1,0} \cdot 2^n + c^{(1)}_{2,1,0} \cdot 1^n + c^{(1)}_{3,1,0} \cdot (-1)^n, \\
f_0(n) &= c^{(2)}_{1,1,0} \cdot 2^n + c^{(2)}_{2,1,0} \cdot 1^n + c^{(2)}_{3,1,0} \cdot (-1)^n, \\
G_0(n) &= c^{(3)}_{1,1,0} \cdot 2^n + c^{(3)}_{2,1,0} \cdot 1^n + c^{(3)}_{2,2,0} \cdot n \cdot 1^n + c^{(3)}_{2,3,0} \cdot n^2 \cdot 1^n + c^{(3)}_{2,4,0} \cdot n^3 \cdot 1^n + c^{(3)}_{3,1,0} \cdot (-1)^n, \\
g_0(n) &= c^{(4)}_{1,1,0} \cdot 2^n + c^{(4)}_{2,1,0} \cdot 1^n + c^{(4)}_{2,2,0} \cdot n \cdot 1^n + c^{(4)}_{2,3,0} \cdot n^2 \cdot 1^n + c^{(4)}_{2,4,0} \cdot n^3 \cdot 1^n + c^{(4)}_{3,1,0} \cdot (-1)^n,
\end{aligned}$$

$$\begin{aligned}
T_0^{(q_1,a,q_0)}(n) &= c^{(5)}_{1,1,0} \cdot 2^n + c^{(5)}_{1,2,0} \cdot n \cdot 2^n + c^{(5)}_{2,1,0} \cdot 1^n + c^{(5)}_{2,2,0} \cdot n \cdot 1^n + c^{(5)}_{3,1,0} \cdot (-1)^n + c^{(5)}_{3,2,0} \cdot n \cdot (-1)^n, \\
t_0^{(q_1,a,q_0)}(n) &= c^{(6)}_{1,1,0} \cdot 2^n + c^{(6)}_{1,2,0} \cdot n \cdot 2^n + c^{(6)}_{2,1,0} \cdot 1^n + c^{(6)}_{2,2,0} \cdot n \cdot 1^n + c^{(6)}_{3,1,0} \cdot (-1)^n + c^{(6)}_{3,2,0} \cdot n \cdot (-1)^n, \\
U_0^{(q_1,a,q_0)}(n) &= c^{(7)}_{1,1,0} \cdot 2^n + c^{(7)}_{1,2,0} \cdot n \cdot 2^n + c^{(7)}_{2,1,0} \cdot 1^n + c^{(7)}_{2,2,0} \cdot n \cdot 1^n + c^{(7)}_{2,3,0} \cdot n^2 \cdot 1^n + c^{(7)}_{2,4,0} \cdot n^3 \cdot 1^n + \\
&\quad + c^{(7)}_{2,5,0} \cdot n^4 \cdot 1^n + c^{(7)}_{3,1,0} \cdot (-1)^n + c^{(7)}_{3,2,0} \cdot n \cdot (-1)^n, \\
u_0^{(q_1,a,q_0)}(n) &= c^{(8)}_{1,1,0} \cdot 2^n + c^{(8)}_{1,2,0} \cdot n \cdot 2^n + c^{(8)}_{2,1,0} \cdot 1^n + c^{(8)}_{2,2,0} \cdot n \cdot 1^n + c^{(8)}_{2,3,0} \cdot n^2 \cdot 1^n + c^{(8)}_{2,4,0} \cdot n^3 \cdot 1^n + \\
&\quad + c^{(8)}_{2,5,0} \cdot n^4 \cdot 1^n + c^{(8)}_{3,1,0} \cdot (-1)^n + c^{(8)}_{3,2,0} \cdot n \cdot (-1)^n,
\end{aligned}$$

$$S_0^{q_0}(n) = c^{(9)}_{1,1,0} \cdot 2^n + c^{(9)}_{1,2,0} \cdot n \cdot 2^n + c^{(9)}_{2,1,0} \cdot 1^n + c^{(9)}_{2,2,0} \cdot n \cdot 1^n + c^{(9)}_{3,1,0} \cdot (-1)^n + c^{(9)}_{3,2,0} \cdot n \cdot (-1)^n,$$

$$\begin{aligned}
s_0^{q_0}(n) &= c_{1,1,0}^{(10)} \cdot 2^n + c_{1,2,0}^{(10)} \cdot n \cdot 2^n + c_{2,1,0}^{(10)} \cdot 1^n + c_{2,2,0}^{(10)} \cdot n \cdot 1^n + c_{3,1,0}^{(10)} \cdot (-1)^n + c_{3,2,0}^{(10)} \cdot n \cdot (-1)^n, \\
V_0^{q_0}(n) &= c_{1,1,0}^{(11)} \cdot 2^n + c_{1,2,0}^{(11)} \cdot n \cdot 2^n + c_{2,1,0}^{(11)} \cdot 1^n + c_{2,2,0}^{(11)} \cdot n \cdot 1^n + c_{2,3,0}^{(11)} \cdot n^2 \cdot 1^n + c_{2,4,0}^{(11)} \cdot n^3 \cdot 1^n + \\
&\quad + c_{2,5,0}^{(11)} \cdot n^4 \cdot 1^n + c_{3,1,0}^{(11)} \cdot (-1)^n + c_{3,2,0}^{(11)} \cdot n \cdot (-1)^n, \\
v_0^{q_0}(n) &= c_{1,1,0}^{(12)} \cdot 2^n + c_{1,2,0}^{(12)} \cdot n \cdot 2^n + c_{2,1,0}^{(12)} \cdot 1^n + c_{2,2,0}^{(12)} \cdot n \cdot 1^n + c_{2,3,0}^{(12)} \cdot n^2 \cdot 1^n + c_{2,4,0}^{(12)} \cdot n^3 \cdot 1^n + \\
&\quad + c_{2,5,0}^{(12)} \cdot n^4 \cdot 1^n + c_{3,1,0}^{(12)} \cdot (-1)^n + c_{3,2,0}^{(12)} \cdot n \cdot (-1)^n,
\end{aligned}$$

for some constants $c_{i,j,0}^{(k)}$ in $\mathbb{C}$. In what follows, we shall determine these constants and thus complete the process of finding closed form solutions for the enumerated quantities.

First of all, let us find a closed form solution for the function $F_0(n)$. The value of $F_0(n)$ is the first component of the vector $\mathbf{F}_n$. From the initial conditions, we have

$$\mathbf{F}_0 = (1, 1, 1)^T.$$

Thus, $F_0(0) = 1$. However, to determine the constants $c_{1,1,0}^{(1)}, c_{2,1,0}^{(1)}$, and $c_{3,1,0}^{(1)}$, we need the values of $F_0(n)$ for $n = 0, 1, 2$. We shall compute the values of $F_0(1)$ and $F_0(2)$ from $\mathbf{F}_1$ and $\mathbf{F}_2$, which we shall compute by a direct left-multiplication of the vector $\mathbf{F}_0$ by the matrix Δ. We shall obtain

$$\begin{aligned}
\mathbf{F}_1 &= \Delta \cdot \mathbf{F}_0 = (2, 1, 2)^T, \\
\mathbf{F}_2 &= \Delta^2 \cdot \mathbf{F}_0 = (3, 2, 4)^T.
\end{aligned}$$

Thus, $F_0(1) = 2$ and $F_0(2) = 3$. The constants $c_{1,1,0}^{(1)}, c_{2,1,0}^{(1)}$, and $c_{3,1,0}^{(1)}$ therefore satisfy the system of linear equations

$$\begin{aligned}
c_{1,1,0}^{(1)} + c_{2,1,0}^{(1)} + c_{3,1,0}^{(1)} &= 1, \\
2 \cdot c_{1,1,0}^{(1)} + c_{2,1,0}^{(1)} - c_{3,1,0}^{(1)} &= 2, \\
4 \cdot c_{1,1,0}^{(1)} + c_{2,1,0}^{(1)} + c_{3,1,0}^{(1)} &= 3,
\end{aligned}$$

i.e., in the matrix-vector form,

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & -1 \\ 4 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} c_{1,1,0}^{(1)} \\ c_{2,1,0}^{(1)} \\ c_{3,1,0}^{(1)} \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

By solving this system, we obtain

$$c_{1,1,0}^{(1)} = \frac{2}{3}, \qquad c_{2,1,0}^{(1)} = \frac{1}{2}, \qquad c_{3,1,0}^{(1)} = -\frac{1}{6}.$$

The closed form solution for the function $F_0(n)$ therefore is

$$F_0(n) = \frac{2}{3} \cdot 2^n + \frac{1}{2} - \frac{1}{6} \cdot (-1)^n.$$

Similarly, we shall determine these constants also for the rest of enumerated functions. For $f_0(n)$, we have

$$\mathbf{f}_0 = (0, 0, 1)^T,$$

from which we get

$$\begin{aligned}
\mathbf{f}_1 &= \Delta \cdot \mathbf{f}_0 = (1, 0, 2)^T, \\
\mathbf{f}_2 &= \Delta^2 \cdot \mathbf{f}_0 = (2, 1, 4)^T.
\end{aligned}$$

The constants $c^{(2)}_{1,1,0}, c^{(2)}_{2,1,0}$, and $c^{(2)}_{3,1,0}$ therefore satisfy the system of linear equations

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & -1 \\ 4 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} c^{(2)}_{1,1,0} \\ c^{(2)}_{2,1,0} \\ c^{(2)}_{3,1,0} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}.$$

By solving this system, we obtain

$$c^{(2)}_{1,1,0} = \frac{2}{3}, \qquad c^{(2)}_{2,1,0} = -\frac{1}{2}, \qquad c^{(2)}_{3,1,0} = -\frac{1}{6}.$$

The closed form solution for the function $f_0(n)$ therefore is

$$f_0(n) = \frac{2}{3} \cdot 2^n - \frac{1}{2} - \frac{1}{6} \cdot (-1)^n.$$

Now, we shall compute $G_0(n)$. From the initial conditions, we have

$$\mathbf{G}_0 = (1,1,1,1,1,1),$$

from which we get

$$\mathbf{G}_1 = \begin{pmatrix} \mathbf{I}_m & \Delta \\ \mathbf{0} & \Delta \end{pmatrix} \cdot \mathbf{G}_0 = (3,2,3,2,1,2)^T,$$
$$\mathbf{G}_2 = \begin{pmatrix} \mathbf{I}_m & \Delta \\ \mathbf{0} & \Delta \end{pmatrix}^2 \cdot \mathbf{G}_0 = (6,4,7,3,2,4)^T,$$
$$\mathbf{G}_3 = \begin{pmatrix} \mathbf{I}_m & \Delta \\ \mathbf{0} & \Delta \end{pmatrix}^3 \cdot \mathbf{G}_0 = (12,7,15,6,3,8)^T,$$
$$\mathbf{G}_4 = \begin{pmatrix} \mathbf{I}_m & \Delta \\ \mathbf{0} & \Delta \end{pmatrix}^4 \cdot \mathbf{G}_0 = (23,13,31,11,6,16)^T,$$
$$\mathbf{G}_5 = \begin{pmatrix} \mathbf{I}_m & \Delta \\ \mathbf{0} & \Delta \end{pmatrix}^5 \cdot \mathbf{G}_0 = (45,24,63,22,11,32)^T.$$

Thus, the constants $c^{(3)}_{1,1,0}, c^{(3)}_{2,1,0}, c^{(3)}_{2,2,0}, c^{(3)}_{2,3,0}, c^{(3)}_{2,4,0}$, and $c^{(3)}_{3,1,0}$ satisfy the following system of linear equations:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 2 & 1 & 1 & 1 & 1 & -1 \\ 4 & 1 & 2 & 4 & 8 & 1 \\ 8 & 1 & 3 & 9 & 27 & -1 \\ 16 & 1 & 4 & 16 & 64 & 1 \\ 32 & 1 & 5 & 25 & 125 & -1 \end{pmatrix} \cdot \begin{pmatrix} c^{(3)}_{1,1,0} \\ c^{(3)}_{2,1,0} \\ c^{(3)}_{2,2,0} \\ c^{(3)}_{2,3,0} \\ c^{(3)}_{2,4,0} \\ c^{(3)}_{3,1,0} \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 6 \\ 12 \\ 23 \\ 45 \end{pmatrix}.$$

By solving this system, we obtain

$$c^{(3)}_{1,1,0} = \frac{4}{3}, \qquad c^{(3)}_{2,1,0} = -\frac{1}{4}, \qquad c^{(3)}_{2,2,0} = \frac{1}{2},$$
$$c^{(3)}_{2,3,0} = 0, \qquad c^{(3)}_{2,4,0} = 0, \qquad c^{(3)}_{3,1,0} = -\frac{1}{12}.$$

Thus, the closed form solution for the function $G_0(n)$ is

$$G_0(n) = \frac{4}{3} \cdot 2^n - \frac{1}{4} + \frac{1}{2} \cdot n - \frac{1}{12} \cdot (-1)^n.$$

Let us now compute the closed form solution for $g_0(n)$. From the initial conditions, we have

$$\mathbf{g}_0 = (0, 0, 1, 0, 0, 1).$$

From that we get

$$\mathbf{g}_1 = \begin{pmatrix} \mathbf{I}_m & \Delta \\ \mathbf{0} & \Delta \end{pmatrix} \cdot \mathbf{g}_0 = (1, 0, 3, 1, 0, 2)^T,$$
$$\mathbf{g}_2 = \begin{pmatrix} \mathbf{I}_m & \Delta \\ \mathbf{0} & \Delta \end{pmatrix}^2 \cdot \mathbf{g}_0 = (3, 1, 7, 2, 1, 4)^T,$$
$$\mathbf{g}_3 = \begin{pmatrix} \mathbf{I}_m & \Delta \\ \mathbf{0} & \Delta \end{pmatrix}^3 \cdot \mathbf{g}_0 = (8, 3, 15, 5, 2, 8)^T,$$
$$\mathbf{g}_4 = \begin{pmatrix} \mathbf{I}_m & \Delta \\ \mathbf{0} & \Delta \end{pmatrix}^4 \cdot \mathbf{g}_0 = (18, 8, 31, 10, 5, 16)^T,$$
$$\mathbf{g}_5 = \begin{pmatrix} \mathbf{I}_m & \Delta \\ \mathbf{0} & \Delta \end{pmatrix}^5 \cdot \mathbf{g}_0 = (39, 18, 63, 21, 10, 32)^T.$$

Thus, for the constants to be determined, the following system of linear equations holds:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 2 & 1 & 1 & 1 & 1 & -1 \\ 4 & 1 & 2 & 4 & 8 & 1 \\ 8 & 1 & 3 & 9 & 27 & -1 \\ 16 & 1 & 4 & 16 & 64 & 1 \\ 32 & 1 & 5 & 25 & 125 & -1 \end{pmatrix} \cdot \begin{pmatrix} c^{(4)}_{1,1,0} \\ c^{(4)}_{2,1,0} \\ c^{(4)}_{2,2,0} \\ c^{(4)}_{2,3,0} \\ c^{(4)}_{2,4,0} \\ c^{(4)}_{3,1,0} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 3 \\ 8 \\ 18 \\ 39 \end{pmatrix}.$$

The solution to this system is

$$c^{(4)}_{1,1,0} = \frac{4}{3}, \qquad c^{(4)}_{2,1,0} = -\frac{5}{4}, \qquad c^{(4)}_{2,2,0} = -\frac{1}{2},$$
$$c^{(4)}_{2,3,0} = 0, \qquad c^{(4)}_{2,4,0} = 0, \qquad c^{(4)}_{3,1,0} = -\frac{1}{12}.$$

The closed form solution for the function $g_0(n)$ therefore is

$$g_0(n) = \frac{4}{3} \cdot 2^n - \frac{5}{4} - \frac{1}{2} \cdot n - \frac{1}{12} \cdot (-1)^n.$$

Up to now, we have found the closed form solutions for functions counting the number of computation paths. Now, we shall turn our attention to functions counting the number of uses of the transition (q_1, a, q_0).

First of all, we shall compute the closed form solution for the function $T_0^{(q_1,a,q_0)}(n)$. From the initial conditions, we have

$$\mathbf{T}_0 = (0, 0, 0, 1, 1, 1),$$

from which we obtain

$$\mathbf{T}_1 = \begin{pmatrix} \Delta & * \\ \mathbf{0} & \Delta \end{pmatrix} \cdot \mathbf{T}_0 = (0, 1, 0, 2, 1, 2)^T,$$
$$\mathbf{T}_2 = \begin{pmatrix} \Delta & * \\ \mathbf{0} & \Delta \end{pmatrix}^2 \cdot \mathbf{T}_0 = (1, 2, 0, 3, 2, 4)^T,$$
$$\mathbf{T}_3 = \begin{pmatrix} \Delta & * \\ \mathbf{0} & \Delta \end{pmatrix}^3 \cdot \mathbf{T}_0 = (2, 4, 0, 6, 3, 8)^T,$$

$$\mathbf{T}_4 = \begin{pmatrix} \Delta & * \\ \mathbf{0} & \Delta \end{pmatrix}^4 \cdot \mathbf{T}_0 = (4,8,0,11,6,16)^T,$$
$$\mathbf{T}_5 = \begin{pmatrix} \Delta & * \\ \mathbf{0} & \Delta \end{pmatrix}^5 \cdot \mathbf{T}_0 = (8,15,0,22,11,32)^T.$$

Of course, in these computations we cannot work with $*$ as a general placeholder for an arbitrary matrix,[5] but we have to work with a concrete matrix that can be determined from the corresponding system of recurrences, i.e., in this case,

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

Thus, for the constants to be determined, the following system of linear equations holds:

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 2 & 2 & 1 & 1 & -1 & -1 \\ 4 & 8 & 1 & 2 & 1 & 2 \\ 8 & 24 & 1 & 3 & -1 & -3 \\ 16 & 64 & 1 & 4 & 1 & 4 \\ 32 & 160 & 1 & 5 & -1 & -5 \end{pmatrix} \cdot \begin{pmatrix} c^{(5)}_{1,1,0} \\ c^{(5)}_{1,2,0} \\ c^{(5)}_{2,1,0} \\ c^{(5)}_{2,2,0} \\ c^{(5)}_{3,1,0} \\ c^{(5)}_{3,2,0} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 2 \\ 4 \\ 8 \end{pmatrix}.$$

By solving this system, we obtain

$$c^{(5)}_{1,1,0} = \frac{2}{9}, \qquad c^{(5)}_{1,2,0} = 0, \qquad c^{(5)}_{2,1,0} = -\frac{1}{2},$$
$$c^{(5)}_{2,2,0} = \frac{1}{4}, \qquad c^{(5)}_{3,1,0} = \frac{5}{18}, \qquad c^{(5)}_{3,2,0} = -\frac{1}{12}.$$

Thus, the closed form solution for the function $T_0^{(q_1,a,q_0)}(n)$ is

$$T_0^{(q_1,a,q_0)}(n) = \frac{2}{9} \cdot 2^n - \frac{1}{2} + \frac{1}{4} \cdot n + \frac{5}{18} \cdot (-1)^n - \frac{1}{12} \cdot n \cdot (-1)^n.$$

Let us now find the closed form solution for the function $t_0^{(q_1,a,q_0)}(n)$. From the initial conditions, we have

$$\mathbf{t}_0 = (0,0,0,0,0,1).$$

From that we obtain

$$\mathbf{t}_1 = \begin{pmatrix} \Delta & * \\ \mathbf{0} & \Delta \end{pmatrix} \cdot \mathbf{t}_0 = (0,0,0,1,0,2)^T,$$
$$\mathbf{t}_2 = \begin{pmatrix} \Delta & * \\ \mathbf{0} & \Delta \end{pmatrix}^2 \cdot \mathbf{t}_0 = (0,1,0,2,1,4)^T,$$
$$\mathbf{t}_3 = \begin{pmatrix} \Delta & * \\ \mathbf{0} & \Delta \end{pmatrix}^3 \cdot \mathbf{t}_0 = (1,2,0,5,2,8)^T,$$
$$\mathbf{t}_4 = \begin{pmatrix} \Delta & * \\ \mathbf{0} & \Delta \end{pmatrix}^4 \cdot \mathbf{t}_0 = (2,6,0,10,5,16)^T,$$

[5] This is one of the rare cases that we are interested in the form of the matrix corresponding to $*$. However, we can avoid this, since the initial values of $T_0^{(q_1,a,q_0)}(n)$ can be computed also by manually examining all computation paths of length n, for initial values of n.

$$\mathbf{t}_5 = \begin{pmatrix} \Delta & * \\ \mathbf{0} & \Delta \end{pmatrix}^5 \cdot \mathbf{t}_0 = (6, 12, 0, 21, 10, 32)^T.$$

Once again,

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

Thus, for the constants to be determined, the following system of linear equations holds:

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 2 & 2 & 1 & 1 & -1 & -1 \\ 4 & 8 & 1 & 2 & 1 & 2 \\ 8 & 24 & 1 & 3 & -1 & -3 \\ 16 & 64 & 1 & 4 & 1 & 4 \\ 32 & 160 & 1 & 5 & -1 & -5 \end{pmatrix} \cdot \begin{pmatrix} c^{(6)}_{1,1,0} \\ c^{(6)}_{1,2,0} \\ c^{(6)}_{2,1,0} \\ c^{(6)}_{2,2,0} \\ c^{(6)}_{3,1,0} \\ c^{(6)}_{3,2,0} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 2 \\ 6 \end{pmatrix}.$$

By solving this system, we obtain

$$\begin{aligned} c^{(6)}_{1,1,0} &= \frac{2}{9}, & c^{(6)}_{1,2,0} &= 0, & c^{(6)}_{2,1,0} &= -\frac{1}{4}, \\ c^{(6)}_{2,2,0} &= -\frac{1}{4}, & c^{(6)}_{3,1,0} &= \frac{1}{36}, & c^{(6)}_{3,2,0} &= -\frac{1}{12}. \end{aligned}$$

Thus, the closed form solution for the function $t_0^{(q_1,a,q_0)}(n)$ is

$$t_0^{(q_1,a,q_0)}(n) = \frac{2}{9} \cdot 2^n - \frac{1}{4} - \frac{1}{4} \cdot n + \frac{1}{36} \cdot (-1)^n - \frac{1}{12} \cdot n \cdot (-1)^n.$$

Now, we shall compute the closed form solution for $U_0^{(q_1,a,q_0)}(n)$. From the initial conditions, we have

$$\mathbf{U}_0 = (0, 0, 0, 0, 0, 0, 1, 1, 1)^T.$$

From that we get

$$\mathbf{U}_1 = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix} \cdot \mathbf{U}_0 = (0, 1, 0, 0, 1, 0, 2, 1, 2)^T,$$

$$\mathbf{U}_2 = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix}^2 \cdot \mathbf{U}_0 = (1, 3, 0, 1, 2, 0, 3, 2, 4)^T,$$

$$\mathbf{U}_3 = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix}^3 \cdot \mathbf{U}_0 = (3, 7, 0, 2, 4, 0, 6, 3, 8)^T,$$

$$\mathbf{U}_4 = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix}^4 \cdot \mathbf{U}_0 = (7, 15, 0, 4, 8, 0, 11, 6, 16)^T,$$

$$\mathbf{U}_5 = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix}^5 \cdot \mathbf{U}_0 = (15, 30, 0, 8, 15, 0, 22, 11, 32)^T,$$

$$\mathbf{U}_6 = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix}^6 \cdot \mathbf{U}_0 = (30, 60, 0, 15, 30, 0, 43, 22, 64)^T,$$

$$\mathbf{U}_7 = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix}^7 \cdot \mathbf{U}_0 = (60, 118, 0, 30, 58, 0, 86, 43, 128)^T,$$
$$\mathbf{U}_8 = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix}^8 \cdot \mathbf{U}_0 = (118, 234, 0, 58, 116, 0, 171, 86, 256)^T.$$

Again,

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

Thus, for the constants to be determined, the following system of linear equations holds:

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 2 & 2 & 1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 4 & 8 & 1 & 2 & 4 & 8 & 16 & 1 & 2 \\ 8 & 24 & 1 & 3 & 9 & 27 & 81 & -1 & -3 \\ 16 & 64 & 1 & 4 & 16 & 64 & 256 & 1 & 4 \\ 32 & 160 & 1 & 5 & 25 & 125 & 625 & -1 & -5 \\ 64 & 384 & 1 & 6 & 36 & 216 & 1296 & 1 & 6 \\ 128 & 896 & 1 & 7 & 49 & 343 & 2401 & -1 & -7 \\ 256 & 2048 & 1 & 8 & 64 & 512 & 4096 & 1 & 8 \end{pmatrix} \cdot \begin{pmatrix} c^{(7)}_{1,1,0} \\ c^{(7)}_{1,2,0} \\ c^{(7)}_{2,1,0} \\ c^{(7)}_{2,2,0} \\ c^{(7)}_{2,3,0} \\ c^{(7)}_{2,4,0} \\ c^{(7)}_{2,5,0} \\ c^{(7)}_{3,1,0} \\ c^{(7)}_{3,2,0} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 3 \\ 7 \\ 15 \\ 30 \\ 60 \\ 118 \end{pmatrix}.$$

By solving this system, we obtain

$$c^{(7)}_{1,1,0} = \frac{4}{9}, \qquad c^{(7)}_{1,2,0} = 0, \qquad c^{(7)}_{2,1,0} = -\frac{9}{16},$$
$$c^{(7)}_{2,2,0} = -\frac{3}{8}, \qquad c^{(7)}_{2,3,0} = \frac{1}{8}, \qquad c^{(7)}_{2,4,0} = 0,$$
$$c^{(7)}_{2,5,0} = 0, \qquad c^{(7)}_{3,1,0} = \frac{17}{144}, \qquad c^{(7)}_{3,2,0} = -\frac{1}{24}.$$

Thus, the closed form solution for the function $U_0^{(q_1,a,q_0)}(n)$ is

$$U_0^{(q_1,a,q_0)}(n) = \frac{4}{9} \cdot 2^n - \frac{9}{16} - \frac{3}{8} \cdot n + \frac{1}{8} \cdot n^2 + \frac{17}{144} \cdot (-1)^n - \frac{1}{24} \cdot n \cdot (-1)^n.$$

Next, we shall compute the closed form solution for $u_0^{(q_1,a,q_0)}(n)$. From the initial conditions, we have

$$\mathbf{u}_0 = (0, 0, 0, 0, 0, 0, 0, 0, 1)^T.$$

From that we obtain

$$\mathbf{u}_1 = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix} \cdot \mathbf{u}_0 = (0, 0, 0, 0, 0, 0, 1, 0, 2)^T,$$
$$\mathbf{u}_2 = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix}^2 \cdot \mathbf{u}_0 = (0, 1, 0, 0, 1, 0, 2, 1, 4)^T,$$
$$\mathbf{u}_3 = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix}^3 \cdot \mathbf{u}_0 = (1, 3, 0, 1, 2, 0, 5, 2, 8)^T,$$

$$\mathbf{u}_4 = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix}^4 \cdot \mathbf{u}_0 = (3, 9, 0, 2, 6, 0, 10, 5, 16)^T,$$

$$\mathbf{u}_5 = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix}^5 \cdot \mathbf{u}_0 = (9, 21, 0, 6, 12, 0, 21, 10, 32)^T,$$

$$\mathbf{u}_6 = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix}^6 \cdot \mathbf{u}_0 = (21, 48, 0, 12, 27, 0, 42, 21, 64)^T,$$

$$\mathbf{u}_7 = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix}^7 \cdot \mathbf{u}_0 = (48, 102, 0, 27, 54, 0, 85, 42, 128)^T,$$

$$\mathbf{u}_8 = \begin{pmatrix} \mathbf{I}_m & \Delta & * \\ \mathbf{0} & \Delta & * \\ \mathbf{0} & \mathbf{0} & \Delta \end{pmatrix}^8 \cdot \mathbf{u}_0 = (102, 214, 0, 54, 112, 0, 170, 85, 256)^T.$$

Again,

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

Thus, for the constants to be determined, the following system of linear equations holds:

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 2 & 2 & 1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 4 & 8 & 1 & 2 & 4 & 8 & 16 & 1 & 2 \\ 8 & 24 & 1 & 3 & 9 & 27 & 81 & -1 & -3 \\ 16 & 64 & 1 & 4 & 16 & 64 & 256 & 1 & 4 \\ 32 & 160 & 1 & 5 & 25 & 125 & 625 & -1 & -5 \\ 64 & 384 & 1 & 6 & 36 & 216 & 1296 & 1 & 6 \\ 128 & 896 & 1 & 7 & 49 & 343 & 2401 & -1 & -7 \\ 256 & 2048 & 1 & 8 & 64 & 512 & 4096 & 1 & 8 \end{pmatrix} \cdot \begin{pmatrix} c_{1,1,0}^{(8)} \\ c_{1,2,0}^{(8)} \\ c_{2,1,0}^{(8)} \\ c_{2,2,0}^{(8)} \\ c_{2,3,0}^{(8)} \\ c_{2,4,0}^{(8)} \\ c_{2,5,0}^{(8)} \\ c_{3,1,0}^{(8)} \\ c_{3,2,0}^{(8)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 3 \\ 9 \\ 21 \\ 48 \\ 102 \end{pmatrix}.$$

By solving this system, we get

$$\begin{aligned} c_{1,1,0}^{(8)} &= \frac{4}{9}, & c_{1,2,0}^{(8)} &= 0, & c_{2,1,0}^{(8)} &= -\frac{7}{16}, \\ c_{2,2,0}^{(8)} &= -\frac{3}{8}, & c_{2,3,0}^{(8)} &= -\frac{1}{8}, & c_{2,4,0}^{(8)} &= 0, \\ c_{2,5,0}^{(8)} &= 0, & c_{3,1,0}^{(8)} &= -\frac{1}{144}, & c_{3,2,0}^{(8)} &= -\frac{1}{24}. \end{aligned}$$

The closed form solution for the function $u_0^{(q_1,a,q_0)}(n)$ therefore is

$$u_0^{(q_1,a,q_0)}(n) = \frac{4}{9} \cdot 2^n - \frac{7}{16} - \frac{3}{8} \cdot n - \frac{1}{8} \cdot n^2 - \frac{1}{144} \cdot (-1)^n - \frac{1}{24} \cdot n \cdot (-1)^n.$$

Finally, we shall find the closed form solutions for functions counting the number of uses of the state q_0. One possibility how to achieve this goal is to find the unknown constant coefficients in exactly the same way as in the previous cases. However, from Theorem 2.1.11, we know the strong relation between functions counting the number of uses of states, and functions counting the number of uses of transitions. That is, according to Theorem 2.1.11, we directly obtain the following equations for unknown constant coefficients:

$$
\begin{aligned}
c^{(9)}_{1,1,0} &= c^{(1)}_{1,1,0} + c^{(5)}_{1,1,0}, & c^{(10)}_{1,1,0} &= c^{(2)}_{1,1,0} + c^{(6)}_{1,1,0}, & c^{(11)}_{1,1,0} &= c^{(3)}_{1,1,0} + c^{(7)}_{1,1,0}, & c^{(12)}_{1,1,0} &= c^{(4)}_{1,1,0} + c^{(8)}_{1,1,0},\\
c^{(9)}_{1,2,0} &= c^{(5)}_{1,2,0}, & c^{(10)}_{1,2,0} &= c^{(6)}_{1,2,0}, & c^{(11)}_{1,2,0} &= c^{(7)}_{1,2,0}, & c^{(12)}_{1,2,0} &= c^{(8)}_{1,2,0},\\
c^{(9)}_{2,1,0} &= c^{(1)}_{2,1,0} + c^{(5)}_{2,1,0}, & c^{(10)}_{2,1,0} &= c^{(2)}_{2,1,0} + c^{(6)}_{2,1,0}, & c^{(11)}_{2,1,0} &= c^{(3)}_{2,1,0} + c^{(7)}_{2,1,0}, & c^{(12)}_{2,1,0} &= c^{(4)}_{2,1,0} + c^{(8)}_{2,1,0},\\
c^{(9)}_{2,2,0} &= c^{(5)}_{2,2,0}, & c^{(10)}_{2,2,0} &= c^{(6)}_{2,2,0}, & c^{(11)}_{2,2,0} &= c^{(3)}_{2,2,0} + c^{(7)}_{2,2,0}, & c^{(12)}_{2,2,0} &= c^{(4)}_{2,2,0} + c^{(8)}_{2,2,0},\\
 & & & & c^{(11)}_{2,3,0} &= c^{(3)}_{2,3,0} + c^{(7)}_{2,3,0}, & c^{(12)}_{2,3,0} &= c^{(4)}_{2,3,0} + c^{(8)}_{2,3,0},\\
 & & & & c^{(11)}_{2,4,0} &= c^{(3)}_{2,4,0} + c^{(7)}_{2,4,0}, & c^{(12)}_{2,4,0} &= c^{(4)}_{2,4,0} + c^{(8)}_{2,4,0},\\
 & & & & c^{(11)}_{2,5,0} &= c^{(7)}_{2,5,0}, & c^{(12)}_{2,5,0} &= c^{(8)}_{2,5,0},\\
c^{(9)}_{3,1,0} &= c^{(1)}_{3,1,0} + c^{(5)}_{3,1,0}, & c^{(10)}_{3,1,0} &= c^{(2)}_{3,1,0} + c^{(6)}_{3,1,0}, & c^{(11)}_{3,1,0} &= c^{(3)}_{3,1,0} + c^{(7)}_{3,1,0}, & c^{(12)}_{3,1,0} &= c^{(4)}_{3,1,0} + c^{(8)}_{3,1,0},\\
c^{(9)}_{3,2,0} &= c^{(5)}_{3,2,0}, & c^{(10)}_{3,2,0} &= c^{(6)}_{3,2,0}, & c^{(11)}_{3,2,0} &= c^{(7)}_{3,2,0}, & c^{(12)}_{3,2,0} &= c^{(8)}_{3,2,0}.
\end{aligned}
$$

Thus, for the functions $S_0^{q_0}(n)$, $s_0^{q_0}(n)$, $V_0^{q_0}(n)$, and $v_0^{q_0}(n)$, we have the following closed form solutions:

$$
\begin{aligned}
S_0^{q_0}(n) &= \frac{8}{9}\cdot 2^n + \frac{1}{4}\cdot n + \frac{1}{9}\cdot(-1)^n - \frac{1}{12}\cdot n\cdot(-1)^n,\\
s_0^{q_0}(n) &= \frac{8}{9}\cdot 2^n - \frac{3}{4} - \frac{1}{4}\cdot n - \frac{5}{36}\cdot(-1)^n - \frac{1}{12}\cdot n\cdot(-1)^n,\\
V_0^{q_0}(n) &= \frac{16}{9}\cdot 2^n - \frac{13}{16} + \frac{1}{8}\cdot n + \frac{1}{8}\cdot n^2 + \frac{5}{144}\cdot(-1)^n - \frac{1}{24}\cdot n\cdot(-1)^n,\\
v_0^{q_0}(n) &= \frac{16}{9}\cdot 2^n - \frac{27}{16} - \frac{7}{8}\cdot n - \frac{1}{8}\cdot n^2 - \frac{13}{144}\cdot(-1)^n - \frac{1}{24}\cdot n\cdot(-1)^n.
\end{aligned}
$$

2.2 Strict $\mathcal{S}$-Equiloadedness

In this section, we shall study strictly $\mathcal{S}$-equiloaded DFA and DFAε, for $\mathcal{S} = \mathcal{C}$ and $\mathcal{S} = \mathcal{A}$. Since acceptance by empty memory does not make sense for deterministic finite automata, we shall not study the case $\mathcal{S} = \mathcal{E}$ in this section.

Although, as we shall see, the families of strictly $\mathcal{C}$-equiloaded and strictly $\mathcal{A}$-equiloaded deterministic finite automata differ, we shall observe that these differences are only minor, and that the corresponding families of languages are in fact the same, both for the case of state-equiloadedness and for the case of transition-equiloadedness.

In Subsection 2.2.1, we shall start our study of strictly $\mathcal{S}$-equiloaded deterministic finite automata by proving characterizations of the families of strictly state-$\mathcal{S}$-equiloaded and strictly transition-$\mathcal{S}$-equiloaded DFA and DFAε, for $\mathcal{S} = \mathcal{C}$ and $\mathcal{S} = \mathcal{A}$. Next, in Subsection 2.2.2, we shall study the relations between the corresponding families of languages. Finally, in Subsection 2.2.3, we shall examine the closure properties of these families of languages.

Some families of strictly $\mathcal{S}$-equiloaded deterministic finite automata and corresponding families of languages have already been studied. In [26] and [27], the family of strictly state-$\mathcal{A}$-equiloaded DFA and the corresponding family of languages $\mathscr{L}_{K-SEQ-DFA}(\mathcal{A})$ have been studied. Moreover, in [25], we have studied the family of strictly transition-$\mathcal{A}$-equiloaded DFA and the corresponding family $\mathscr{L}_{\delta-SEQ-DFA}(\mathcal{A})$. All of the results on these families of automata and languages, presented in this section, have been already obtained in these earlier works. However, strict $\mathcal{C}$-equiloadedness for DFA and DFAε, as well as strict $\mathcal{S}$-equiloadedness for DFAε have not been studied yet. Thus, all of the results on the corresponding families of automata and languages, presented in this section, are new.

2.2.1 Characterizations of Strict $\mathcal{S}$-Equiloadedness for $\mathcal{S} = \mathcal{C}$ and $\mathcal{S} = \mathcal{A}$

In this subsection, we shall prove the characterizations of families of strictly state-$\mathcal{S}$-equiloaded and strictly transition-$\mathcal{S}$-equiloaded DFA and DFAε, for $\mathcal{S} = \mathcal{A}$ and $\mathcal{S} = \mathcal{C}$. The characterization of strictly state-$\mathcal{A}$-equiloaded DFA is due to [26] and [27], and we have proved the characterization of strictly transition-$\mathcal{A}$-equiloaded DFA in [25]. The remaining characterizations are new. However, we shall state all of these characterizations in two theorems.

Theorem 2.2.1 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε.

a) A is strictly state-$\mathcal{C}$-equiloaded, if and only if its graphical representation either does not contain any reachable directed cycle, or is a directed multicycle through all states.

b) A is strictly state-$\mathcal{A}$-equiloaded, if and only if its graphical representation either does not contain any reachable directed cycle from which some accepting state is reachable,[6] or is a directed multicycle through all states.

Proof. Let us first prove the left-to-right implication. Let the automaton A be strictly state-$\mathcal{C}$-equiloaded, i.e., let the inequality

$$|\#[p,\gamma] - \#[q,\gamma]| \leq k \tag{2.4}$$

hold for some k in $\mathbb{N}$, every computation path γ in $\mathrm{Comp}(A)$ and every two states p, q in K. For the purpose of contradiction, let us suppose that the graphical representation contains a reachable directed cycle, and at the same time, is not a directed multicycle through all states.

This clearly implies that states p, q in K exist, such that for some words u, v in Σ^*,

$$(q_0, uv) \vdash^* (p, v) \vdash^+ (p, \varepsilon),$$

where the state q is not used in the computation

$$(p, v) \vdash^+ (p, \varepsilon)$$

and is used at most once in the computation

$$(q_0, uv) \vdash^* (p, v).$$

Now, let us consider the word uv^{k+2} and let us denote the corresponding computation path by γ. Then it is clear that

$$|\#[p,\gamma] - \#[q,\gamma]| \geq k + 1,$$

which contradicts (2.4).

Similarly, let the automaton A be strictly state-$\mathcal{A}$-equiloaded, i.e., let the inequality (2.4) hold for some k in $\mathbb{N}$, every accepting computation path γ in $\mathrm{Acc}(A)$ and every two states p, q in K. Let us suppose that the graphical representation contains a reachable directed cycle from which some accepting state is reachable, and at the same time, is not a directed multicycle through all states.

This implies that for some states p, q in K, accepting state q_F in F and words u, v, w in Σ^*,

$$(q_0, uvw) \vdash^* (p, vw) \vdash^+ (p, w) \vdash^* (q_F, \varepsilon),$$

where the state q is not used in the computation

$$(p, vw) \vdash^+ (p, w)$$

[6] This means that the accepted language $L(A)$ is finite, and at the same time, there is no state q in K, such that $(q, w) \vdash^+ (q, w) \vdash^* (q_F, \varepsilon)$ for some w in Σ^* and q_F in F (i.e., there is no ε-cycle from which an accepting state is reachable). Thus, for DFA, this condition is equivalent to the finiteness of the accepted language $L(A)$.

and is used at most once in the computations

$$(q_0, uvw) \vdash^* (p, vw)$$

and

$$(p, w) \vdash^* (q_F, \varepsilon).$$

Let us consider the word $uv^{k+3}w$, and let us denote the corresponding accepting computation path by γ. Then,

$$|\#[p,\gamma] - \#[q,\gamma]| \geq k+1,$$

which contradicts (2.4) that is supposed to hold for γ.

Now, let us prove the right-to-left implication. If the graphical representation of the automaton A does not contain any reachable directed cycle, then each state of the automaton A can be used at most once in a given computation path. That is,

$$0 \leq \#[q,\gamma] \leq 1$$

for all computation paths γ and each state q in K. Thus, for every two states p, q in K and all computation paths γ,

$$|\#[p,\gamma] - \#[q,\gamma]| \leq 1,$$

and the automaton A is strictly state-$\mathcal{C}$-equiloaded.

Now, let us suppose that the graphical representation of the automaton A is a directed multicycle through all states. Then it can be clearly seen that for every two states p, q in K and all computation paths γ,

$$|\#[p,\gamma] - \#[q,\gamma]| \leq 1.$$

Thus, the automaton A is strictly state-$\mathcal{C}$-equiloaded.

Similarly, if the graphical representation of the automaton A does not contain any reachable directed cycle from which some accepting state is reachable, then each state of the automaton A can be used at most once in a given accepting computation path. That is,

$$0 \leq \#[q,\gamma] \leq 1$$

for all accepting computation paths γ and each state q in K. Thus, for each pair of states p, q in K and all accepting computation paths γ,

$$|\#[p,\gamma] - \#[q,\gamma]| \leq 1,$$

and the automaton A is strictly state-$\mathcal{A}$-equiloaded.

Now, let us suppose that the graphical representation of the automaton A is a directed multicycle through all states. Then, we have already proved that the automaton A is strictly state-$\mathcal{C}$-equiloaded, and thus, clearly, also strictly state-$\mathcal{A}$-equiloaded. □

Theorem 2.2.2 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with connected graphical representation.[7]

a) A is strictly transition-$\mathcal{C}$-equiloaded, if and only if its graphical representation either does not contain any reachable directed cycle, or is a directed cycle through all states.

b) A is strictly transition-$\mathcal{A}$-equiloaded, if and only if its graphical representation either does not contain any reachable directed cycle from which some accepting state is reachable, or is a directed cycle through all states.

[7] This is only a minor technical assumption. If this assumption had been omitted, the rest of the characterization would have not been true for automata that, aside a connected component containing the initial state, have some number of other connected components, each consisting of one isolated state. However, to every deterministic finite automaton it is easy to find an equivalent automaton with connected graphical representation – it suffices to delete all connected components not containing the initial state.

Proof. The proof of this theorem will be analogous to the proof of Theorem 2.2.1. First, let us prove the left-to-right implication. Let the automaton A be strictly transition-$\mathcal{C}$-equiloaded, i.e., let the inequality

$$|\#[e,\gamma] - \#[f,\gamma]| \leq k \tag{2.5}$$

hold for some k in $\mathbb{N}$, every computation path γ in $\mathrm{Comp}(A)$ and each pair of transitions e, f in D. For the purpose of contradiction, let us suppose that the graphical representation contains a reachable directed cycle, and at the same time, is not a directed cycle through all states.

As in the proof of Theorem 2.2.1, this implies that a computation path γ in $\mathrm{Comp}(A)$ exists, such that some transition f in D is used at most once in γ and some other transition e in D is used at least $k+2$ times in γ. That yields an obvious contradiction to (2.5).

For the case of A being strictly transition-$\mathcal{A}$-equiloaded, the inequality (2.5) is required to hold for some k in $\mathbb{N}$, every accepting computation path γ in $\mathrm{Acc}(A)$ and each pair of transitions e, f in D. Let us suppose that the graphical representation contains a reachable directed cycle from which some accepting state is reachable, and at the same time, is not a directed cycle through all states.

Again, similarly as in the proof of Theorem 2.2.1, this implies that an accepting computation path γ in $\mathrm{Acc}(A)$ exists, such that some transition f in D is used at most twice in γ and some other transition e in D is used at least $k+3$ times in γ. Again, this contradicts (2.5) that is required to hold for γ.

Now, let us prove the right-to-left implication. If the graphical representation of the automaton A does not contain any reachable directed cycle, then each transition of the automaton A can be used at most once in a given computation path in $\mathrm{Comp}(A)$. That is,

$$0 \leq \#[e,\gamma] \leq 1,$$

for each transition e in D and every computation path γ in $\mathrm{Comp}(A)$. That is,

$$|\#[e,\gamma] - \#[f,\gamma]| \leq 1$$

for each pair of transitions e, f in D and every computation path γ in $\mathrm{Comp}(A)$. In other words, the automaton A is strictly transition-$\mathcal{C}$-equiloaded.

If the graphical representation of the automaton A does not contain any reachable directed cycle from which some accepting state is reachable, then the reasoning above applies to all accepting computation paths γ in $\mathrm{Acc}(A)$, and the automaton A is strictly transition-$\mathcal{A}$-equiloaded.

Finally, if the graphical representation of the automaton A is a directed cycle through all states, it is obvious that the inequality

$$|\#[e,\gamma] - \#[f,\gamma]| \leq 1$$

holds for every two transitions e, f in D and all computation paths γ in $\mathrm{Comp}(A)$ (and thus also all accepting computation paths γ in $\mathrm{Acc}(A)$), and the automaton A is strictly transition-$\mathcal{C}$-equiloaded, as well as strictly transition-$\mathcal{A}$-equiloaded. We have proved both implications for both claims. Thus, the theorem is proved. □

Finally in this subsection, we shall state two theorems and one corollary concerning strictly $\mathcal{S}$-equiloaded DFA (i.e., deterministic finite automata without ε-transitions) that have been proved for $\mathcal{S} = \mathcal{A}$ in [26], [27] (Theorem 2.2.3), and [25] (Theorem 2.2.4 and Corollary 2.2.5). However, since the proof for the case $\mathcal{S} = \mathcal{C}$ is analogous to the proof for the case $\mathcal{S} = \mathcal{A}$, we shall omit proofs of these statements.

Theorem 2.2.3 Let L in $\mathscr{R}$ be a regular language, and $\mathcal{S}$ be in $\{\mathcal{C}, \mathcal{A}\}$. The language L is a strictly state-$\mathcal{S}$-equiloaded DFA-language, if and only if the minimal DFA accepting L is strictly state-$\mathcal{S}$-equiloaded.

Theorem 2.2.4 Let L in $\mathscr{R}$, $L \subseteq \Sigma^*$, be a regular language, and $\mathcal{S}$ be in $\{\mathcal{C}, \mathcal{A}\}$. The language L is a strictly transition-$\mathcal{S}$-equiloaded DFA-language, if and only if L is finite, or if

$$L = \{u_1 u_2 \dots u_n v\}^* \{u_1, u_1 u_2, \dots, u_1 u_2 \dots u_n\}$$

for some fixed n in $\mathbb{N}$, $u_1, u_2, \dots, u_n$ in Σ^* and v in Σ^+.

Corollary 2.2.5 Let $\mathcal{S}$ be in $\{\mathcal{C}, \mathcal{A}\}$, and L in $\mathscr{L}_{\delta-SEQ-DFA}(\mathcal{S})$ be an infinite language. Let u and v be words in L, such that $|u| \leq |v|$. Then u is a prefix of v.

2.2.2 Relations between the Families of Strictly $\mathcal{S}$-Equiloaded Languages

In this subsection, we shall prove several relations between the families of strictly $\mathcal{S}$-equiloaded DFA-languages and DFAε-languages. We shall observe that some of these families are in fact the same.

Theorem 2.2.6 The following identities hold:

1. $\mathscr{L}_{K-SEQ-DFA}(\mathcal{C}) = \mathscr{L}_{K-SEQ-DFA}(\mathcal{A}) =: \mathscr{L}_{K-SEQ-DFA}$,
2. $\mathscr{L}_{K-SEQ-DFA\varepsilon}(\mathcal{C}) = \mathscr{L}_{K-SEQ-DFA\varepsilon}(\mathcal{A}) =: \mathscr{L}_{K-SEQ-DFA\varepsilon}$,
3. $\mathscr{L}_{\delta-SEQ-DFA}(\mathcal{C}) = \mathscr{L}_{\delta-SEQ-DFA}(\mathcal{A}) =: \mathscr{L}_{\delta-SEQ-DFA}$,
4. $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}(\mathcal{C}) = \mathscr{L}_{\delta-SEQ-DFA\varepsilon}(\mathcal{A}) =: \mathscr{L}_{\delta-SEQ-DFA\varepsilon}$.

Proof. The theorem is a direct consequence of Theorem 2.2.1, Theorem 2.2.2, and the obvious fact that to each deterministic finite automaton A (with or without ε-transitions) without a reachable directed cycle in the graphical representation, from which some accepting state is reachable, an equivalent deterministic finite automaton A' exists, such that there is not any reachable directed cycle in its graphical representation: it clearly suffices to delete all states of the automaton A that belong to some reachable directed cycle – since there is not any accepting state reachable from these states in A, the resulting automaton A' would be clearly equivalent to A. $\square$

Thus, we have proved that, in terms of families of languages, strict $\mathcal{C}$-equiloadedness and strict $\mathcal{A}$-equiloadedness are equivalent for DFA and DFAε. However, let us note that these concepts are not equivalent in terms of families of automata. It is obviously possible to construct a deterministic finite automaton that is strictly $\mathcal{A}$-equiloaded, but not strictly $\mathcal{C}$-equiloaded (both for states and for transitions). The family of strictly $\mathcal{A}$-equiloaded DFA (DFAε) is a proper superset of the family of strictly $\mathcal{C}$-equiloaded DFA (DFAε).

From this observation, it is also possible to conclude that, in terms of families of languages, strict $\mathcal{C}$-equiloadedness and strict $\mathcal{A}$-equiloadedness are not equivalent for general families of automata that can be viewed as a special case of ADA. It suffices to consider a family x of automata consisting of a single deterministic finite automaton that is strictly state-$\mathcal{A}$-equiloaded (strictly transition-$\mathcal{A}$-equiloaded), but not strictly state-$\mathcal{C}$-equiloaded (strictly transition-$\mathcal{C}$-equiloaded). Then, the family of languages $\mathscr{L}_{K-SEQ-x}(\mathcal{A})$ ($\mathscr{L}_{\delta-SEQ-x}(\mathcal{A})$) consists of one single regular language, while the family $\mathscr{L}_{K-SEQ-x}(\mathcal{C})$ ($\mathscr{L}_{\delta-SEQ-x}(\mathcal{C})$) is empty.

Remark 2.2.7 Henceforth, in the context of Theorem 2.2.6, we shall call strictly state-$\mathcal{S}$-equiloaded (strictly transition-$\mathcal{S}$-equiloaded) DFA-languages (DFAε-languages), for $\mathcal{S} = \mathcal{A}$ or $\mathcal{S} = \mathcal{C}$, strictly state-equiloaded (strictly transition-equiloaded) DFA-languages (DFAε-languages).

Theorem 2.2.8 The following strict inclusions hold:

1. $\mathscr{L}_{K-SEQ-DFA} \subsetneq \mathscr{L}_{K-SEQ-DFA\varepsilon}$,
2. $\mathscr{L}_{\delta-SEQ-DFA} \subsetneq \mathscr{L}_{\delta-SEQ-DFA\varepsilon}$.

Proof. The claims $\mathscr{L}_{K-SEQ-DFA} \subseteq \mathscr{L}_{K-SEQ-DFA\varepsilon}$ and $\mathscr{L}_{\delta-SEQ-DFA} \subseteq \mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ are obvious, since every DFA is also a DFAε.

We shall prove that these inclusions are proper. Let us consider the language $L = \{a\}^+$. We shall construct a DFAε $A = (K, \Sigma, \delta, q_0, F)$, such that $L(A) = L$. Let us define the automaton A as follows: $K = \{q_0, q_1\}$, $\Sigma = \{a\}$, $F = \{q_1\}$, and

$$\delta(q_0, a) = q_1,$$
$$\delta(q_1, \varepsilon) = q_0.$$

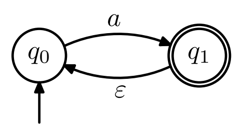


Figure 2.2: The deterministic finite automaton A with ε-transitions, accepting the language $L = \{a\}^+$.

The automaton A is strictly state-$\mathcal{S}$-equiloaded for both $\mathcal{S} = \mathcal{C}$ and $\mathcal{S} = \mathcal{A}$ (however, the equiloadedness of the automaton for one of these $\mathcal{S}$ would suffice), since its graphical representation is a directed multicycle through all states. Moreover, the graphical representation of the automaton A is in fact a directed cycle, and thus the automaton is also strictly transition-$\mathcal{S}$-equiloaded for $\mathcal{S}$ in $\{\mathcal{C}, \mathcal{A}\}$. The claim $L(A) = L$ is considered to be obvious. Thus, L is both in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ and in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$.

However, by applying Theorem 2.2.3, it is easy to prove that L is neither in $\mathscr{L}_{K-SEQ-DFA}$, nor in $\mathscr{L}_{\delta-SEQ-DFA}$ (the minimal DFA accepting L does not satisfy any of the presented characterizations of strict $\mathcal{S}$-equiloadedness). The theorem is proved. $\square$

Theorem 2.2.9 The following strict inclusions hold:

1. $\mathscr{L}_{\delta-SEQ-DFA} \subsetneq \mathscr{L}_{K-SEQ-DFA}$,
2. $\mathscr{L}_{\delta-SEQ-DFA\varepsilon} \subsetneq \mathscr{L}_{K-SEQ-DFA\varepsilon}$.

Proof. The correctness of the inclusions is obvious, since every directed cycle is at the same time a directed multicycle (and thus, by the characterizations given in Theorem 2.2.1 and in Theorem 2.2.2, every strictly transition-$\mathcal{S}$-equiloaded DFA(ε) is strictly state-$\mathcal{S}$-equiloaded as well).

Let us prove that these inclusions are proper and consider the language $L = \{a, b\}^*$. It is obvious that L is in $\mathscr{L}_{K-SEQ-DFA}$ (since it can be accepted by a DFA consisting of one single state), and thus, by Theorem 2.2.8, also in $\mathscr{L}_{K-SEQ-DFA}$. We shall prove that L is not in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ (and, by Theorem 2.2.8, neither in $\mathscr{L}_{\delta-SEQ-DFA}$).

For the purpose of contradiction, let us suppose that a strictly transition-$\mathcal{A}$-equiloaded DFAε $A = (K, \Sigma, \delta, q_0, F)$ exists, such that $L(A) = L$. Since the language L is infinite, the number of accepting computation paths of the automaton A has to be infinite as well. Thus, according to the characterization from Theorem 2.2.2, the graphical representation of the automaton A is a directed cycle through all states. Moreover, F is a nonempty set.

Let q in K be the first state after q_0 in the direction of the directed cycle, such that there is a transition on a character from Σ (that is, not an ε-transition), leading from q (such a state has to exist, since the language $L(A)$ is nonempty). Since the graphical representation of the automaton A is a directed cycle through all states, there has to be *exactly one* transition leading from q. But if this transition is on a, then every word from $L(A)$ has to begin with a, and if this transition is on b, then every word from $L(A)$ has to begin with b. In both cases, this is a contradiction with the assumption $L(A) = L$. $\square$

2.2.3 Closure Properties

In this subsection, we shall examine the closure properties of the families of strictly $\mathcal{S}$-equiloaded DFA-languages and DFAε-languages. We shall prove only the closure properties that have not been examined yet. The proofs of the closure properties that have already been examined, can be found in [26], [27], and [25].

Closure Properties of the Family $\mathscr{L}_{K-SEQ-DFA}$

Closure properties of the family of languages $\mathscr{L}_{K-SEQ-DFA}(\mathcal{A}) = \mathscr{L}_{K-SEQ-DFA}$ have been proved in [26] and [27]. Thus, we shall omit the proof of the following theorem.

Theorem 2.2.10 The family $\mathscr{L}_{K-SEQ-DFA}$ is closed under intersection and not closed under concatenation, union, complementation, Kleene star, Kleene plus, reversal, homomorphism, and inverse homomorphism.

Closure Properties of the Family $\mathscr{L}_{K-SEQ-DFA\varepsilon}$

Closure properties of the family $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ have not been studied yet. Thus, we shall include also proofs of the following theorems.

Theorem 2.2.11 The family $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ is not closed under concatenation.

Proof. Let us consider languages $L_1 = \{a\}^*$ and $L_2 = \{b\}$. For both of these languages, it is clearly possible to construct a strictly state-$\mathcal{A}$-equiloaded automaton accepting that language, and thus, both of these languages are in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. Now, let us consider the language

$$L_1 \cdot L_2 = \{a\}^* \cdot \{b\} = \{a^i b \mid i \in \mathbb{N}\}.$$

We shall prove that $L_1 \cdot L_2$ is not in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. For the purpose of contradiction, let us suppose that a strictly state-$\mathcal{A}$-equiloaded DFAε A exists, such that $L(A) = L_1 \cdot L_2$. The language $L_1 \cdot L_2$ is infinite and hence, the set $\mathrm{Acc}(A)$ is infinite. Thus, by Theorem 2.2.1, the graphical representation of the automaton A is a directed multicycle through all states. Thus, the graphical representation of the automaton A is strongly connected. Moreover, the automaton A has to have at least one accepting state. Since there are words in the language $L_1 \cdot L_2$ containing the character b, the automaton A has to have at least one transition e on the character b. Then, the strong connectedness of the graphical representation together with the existence of at least one accepting state implies that an accepting computation path of the automaton A exists, such that the transition e is used at least twice in this computation. That is, the language $L_1 \cdot L_2$ contains at least one word with at least two occurrences of the character b, and this clearly contradicts the assumption $L(A) = L_1 \cdot L_2$. □

Theorem 2.2.12 The family $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ is not closed under union.

Proof. Again, let us consider languages $L_1 = \{a\}^*$ and $L_2 = \{b\}$, both in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. We shall prove that the language

$$L_1 \cup L_2 = \{a\}^* \cup \{b\}$$

is not in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. By contradiction. Let us suppose that a strictly state-$\mathcal{A}$-equiloaded DFAε A exists, such that $L(A) = L_1 \cup L_2$. Since the language $L_1 \cup L_2$ is infinite, it follows from Theorem 2.2.1 that the graphical representation of the automaton A is a directed multicycle through all states, and thus, is strongly connected. Moreover, the automaton A has at least one transition e on the character b, since otherwise it could not accept the language $L_1 \cup L_2$. Further, since the automaton A accepts a nonempty language, it has at least one accepting state. From these observations, it follows that at least one accepting computation path γ of the automaton A exists, such that the transition e is used at least twice in γ. Thus, obviously, $L(A) \neq L_1 \cup L_2$ and that is a contradiction. $\square$

To prove that the family $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ is closed under intersection, we shall need the following lemma.

Lemma 2.2.13 Let L be an infinite language in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. Then a strictly state-$\mathcal{A}$-equiloaded DFA$\varepsilon$ $A = (K, \Sigma, \delta, q_0, F)$ exists, in the following normal form:

(i) The automaton A has *exactly one* ε-transition e.

(ii) The transition e leads to the state q_0.

Proof. Since L is an infinite language in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$, it follows from Theorem 2.2.1 that a DFAε $A' = (K', \Sigma', \delta', q'_0, F')$ accepting L exists, such that its graphical representation is a directed multicycle through all states. We shall prove by induction on $m = |K'|$ that the automaton A' can be transformed to an equivalent automaton A satisfying (i) and (ii).

1. Let $m = 1$. From the definition of DFAε, the transitions leading from (and to) the only state q'_0 are either all on some character in Σ, or there is only one ε-transition. If the latter was true, then the language L would be finite, and we would get a contradiction. Thus, there is not any ε-transition in A.

   Moreover, the state q'_0 has to be accepting, since otherwise the language L would be empty, and thus finite. Thus, it is obviously possible to define the automaton $A = (K, \Sigma, \delta, q_0, F)$ as follows: $K = \{q_0, q_1\}$, $\Sigma = \Sigma'$, $F = \{q_0\}$, and

   $$\begin{aligned} &\delta(q_0, c) = q_1, \qquad \forall c \in \Sigma : \delta'(q'_0, c) = q'_0, \\ &\delta(q_1, \varepsilon) = q_0. \end{aligned}$$

   It is clear that $L(A) = L(A') = L$, and that the conditions (i) and (ii) are satisfied by A.

2. Let the property hold for all $m \leq k$ for some k in $\mathbb{N}$. We shall show that it holds also for $m = k + 1$.

   If the automaton A' does not have any ε-transition, then it can be inserted in a manner similar to basis of the induction. More formally, it is possible to define the automaton $A = (K, \Sigma, \delta, q_0, F)$ as follows: $K = K' \cup \{q\}$, where q is a new state, $\Sigma = \Sigma'$, $q_0 = q'_0$, $F = F'$, and

   $$\begin{aligned} &\delta(p, c) = \delta'(p, c), \qquad \forall p \in K' \forall c \in \Sigma : \delta'(p, c) = r \neq q'_0, \\ &\delta(p, c) = q, \qquad \forall p \in K' \forall c \in \Sigma : \delta'(p, c) = q'_0, \\ &\delta(q, \varepsilon) = q'_0. \end{aligned}$$

   It is obvious that $L(A) = L(A') = L$, and that the conditions (i) and (ii) are satisfied by A.

   Now, since the graphical representation of the automaton A' has a form of a directed multicycle through all states, only two possibilities are left. The first is that the conditions (i) and (ii) are satisfied. In that case, the induction step is proved. The second possibility is that there is an ε-transition $e = (p, \varepsilon, q)$ that does not lead to q_0. Then, let us define a DFAε $A'' = (K'', \Sigma'', \delta'', q''_0, F'')$ as follows: $K'' = K' - \{q\}$, $\Sigma'' = \Sigma'$,

   $$\begin{aligned} &\delta''(r, c) = \delta'(r, c), \qquad \forall r \in K'' - \{p\} \forall c \in \Sigma'' \cup \{\varepsilon\}, \\ &\delta''(p, c) = \delta'(q, c), \qquad \forall c \in \Sigma'' \cup \{\varepsilon\}, \end{aligned}$$

   $q''_0 = q'_0$, and finally,

   $$F'' = \begin{cases} (F' - \{q\}) \cup \{p\} & \text{if } q \text{ is in } F' \\ F' - \{q\} & \text{if } q \text{ is not in } F' \end{cases}.$$

   Clearly, $L(A'') = L(A')$. Moreover, $|K''| \leq k$, and thus, the induction hypothesis applies. The induction step is therefore proved.

Thus, the lemma is proved. However, let us note that in general, it is not possible to get completely rid of ε-transitions while preserving strict state-equiloadedness: let w in Σ^+ be a word, such that

$$(q_0, w) \vdash^* (q_0, \varepsilon).$$

Now, if w is in L and, at the same time, ε is not in L, then the ε-transition leading to q_0 is necessary. After all, this is the reason for which $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ is a *proper* superset of $\mathscr{L}_{K-SEQ-DFA}$. □

Theorem 2.2.14 The family $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ is closed under intersection.

Proof. Let L_1, L_2 be languages in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. If at least one of these languages is finite, then also their intersection $L_1 \cap L_2$ is finite, and thus in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. The same is true, if L_1, L_2 are infinite languages with finite intersection.

Now, let both of the languages L_1, L_2 be infinite, and let also their intersection $L_1 \cap L_2$ be infinite. Let $A_1 = (K_1, \Sigma_1, \delta_1, p_0, F_1)$ and $A_2 = (K_2, \Sigma_2, \delta_2, r_0, F_2)$ be strictly state-$\mathcal{A}$-equiloaded DFAε in the normal form from Lemma 2.2.13, such that $L(A_1) = L_1$ and $L(A_2) = L_2$.

Let us denote the states of the automaton A_1 in the direction of the multicycle by $p_0, p_1, \ldots, p_m$, where m is in $\mathbb{N}$. Similarly, let us denote the states of the automaton A_2 in the direction of the multicycle by $r_0, r_1, \ldots, r_n$, where n is in $\mathbb{N}$. Without loss of generality, let p_m be in F_1 whenever p_0 is in F_1 and r_n be in F_2 whenever r_0 is in F_2. Let k be the least common multiple of the numbers m and n.

We shall define a DFAε $A = (K, \Sigma, \delta, q_0, F)$ accepting $L_1 \cap L_2$ as follows: $K = \{q_0, q_1, \ldots, q_k\}$, $\Sigma = \Sigma_1 \cap \Sigma_2$. The set of accepting states F shall be defined by

$$q_0 \in F \iff p_0 \in F_1 \ \wedge \ r_0 \in F_2,$$

and for i in $\{1, 2, \ldots, k\}$,

$$q_i \in F \iff p_{[(i-1) \bmod m]+1} \in F_1 \ \wedge \ r_{[(i-1) \bmod n]+1} \in F_2.$$

Finally, the transition function δ shall be defined for q_i with i in $\{0, 1, \ldots, k-1\}$ by

$$\delta(q_i, c) = q_{i+1} \iff \delta_1(p_{i \bmod m}) = p_{[i \bmod m]+1} \ \wedge \ \delta_2(r_{i \bmod n}) = r_{[i \bmod n]+1}$$

and for q_k as

$$\delta(q_k, \varepsilon) = q_0.$$

Clearly, $L(A) = L_1 \cap L_2$. Since the intersection $L_1 \cap L_2$ is infinite, the graphical representation of the automaton A is a directed multicycle through all states. Thus, the automaton A is strictly state-$\mathcal{A}$-equiloaded, and this implies that $L(A) = L_1 \cap L_2$ is in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. □

Theorem 2.2.15 The family $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ is not closed under complementation.

Proof. Let us consider a language $L = \{\varepsilon, a\}$. The language L is finite, and thus in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. We shall prove that $L^C = \{a^k \mid k \geq 2\}$ is not in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$.

For the purpose of contradiction, let us suppose that L^C is in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. Then, a strictly state-$\mathcal{A}$-equiloaded DFAε $A = (K, \Sigma, \delta, q_0, F)$ exists, such that $L(A) = L^C$. Since the language L^C is infinite, the graphical representation of the automaton A is a directed multicycle through all states and A has at least one accepting state.

We can clearly assume that $\Sigma = \{a\}$. This implies that we can without loss of generality suppose that the graphical representation of the automaton A is a directed *cycle* through all states. Let $w = a^j$ be a shortest word in Σ^*, such that

$$(q_0, w) \vdash^* (q_0, \varepsilon).$$

Clearly, $j \geq 2$, since otherwise it would be possible to reach all states of the automaton A by reading ε or a, and that contradicts the assumption that A has at least one accepting state.

Obviously, there is not any accepting state q, such that $(q_0, a) \vdash^* (q, \varepsilon)$. This implies that there is not any accepting state p, such that $(q_0, a^{j+1}) \vdash^* (p, \varepsilon)$. Thus, a^{j+1} is not in $L(A)$. However, since $j \geq 2$, this contradicts the assumption that $L(A) = L^C$. □

Theorem 2.2.16 The family $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ is not closed under closure.

Proof. Let us consider a language $L = \{aa, bb\}$. Since the language L is finite, L is a member of the family $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. We shall prove that the language $L^* = \{aa, bb\}^*$ is not in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$.

By contradiction. Let us suppose that L^* is in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. Then, a strictly state-$\mathcal{A}$-equiloaded DFAε $A = (K, \Sigma, \delta, q_0, F)$ exists, such that $L(A) = L$. Since the language L^* is infinite, the graphical representation of the automaton A is a directed multicycle through all states and the automaton A has at least one accepting state. Moreover, the automaton A has to have at least one state p in K, such that a transition (p, b, p') on the character b exists in D – otherwise, it would be $L(A) \neq L^*$.

Further, we can observe that if a transition e on a leads between some two states, then also some transition on b leads between these two states, and vice versa. Otherwise, for some n_0 in $\mathbb{N}$, all words in $L(A)$ of length $n \geq n_0$ would have at least one occurrence of the character a (resp. b). However, this contradicts $L(A) = L^*$. Thus, we have proved that each state of the automaton A can be reached from each other state without reading the character a, and also, by some in general different sequence of transitions, without reading the character b.

We shall prove that nonnegative integers i, j in $\mathbb{N}$ exist, such that a^iba^j is in $L(A)$, and thus, $L(A) \neq L^*$. Since each state of the automaton A is reachable from each other state without reading the character b, a nonnegative integer i in $\mathbb{N}$ exists, such that

$$(q_0, a^i) \vdash^* (p, \varepsilon),$$

and a nonnegative integer j in $\mathbb{N}$ exists, such that

$$(p', a^j) \vdash^* (q, \varepsilon),$$

where q in F is an arbitrary accepting state. Thus,

$$(q_0, a^iba^j) \vdash^* (p, ba^j) \vdash^* (p', a^j) \vdash^* (q, \varepsilon),$$

i.e., a^iba^j is in $L(A)$. This implies $L(A) \neq L^*$, and that is a contradiction. □

Theorem 2.2.17 The family $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ is not closed under positive closure.

Proof. Exactly the same counterexample and exactly the same argumentation can be used, as in the proof of Theorem 2.2.16. □

Theorem 2.2.18 The family $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ is not closed under reversal.

Proof. Let us consider a language $L = \{ab\}^*\{\varepsilon, a\}$. Clearly, L is in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. We shall prove that L^R is not in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$.

For the purpose of contradiction, let us suppose that L^R is in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. Then, a strictly state-$\mathcal{A}$-equiloaded DFAε $A = (K, \Sigma, \delta, q_0, F)$ exists, such that $L(A) = L^R$. Since the language L^R is infinite, the graphical representation of the automaton A is a directed multicycle through all states. For the same reason, the set of accepting states F is nonempty.

Clearly, a state p in K exists, such that $(q_0, a) \vdash^* (p, \varepsilon)$. Further, a state q in K exists, such that $(q_0, b) \vdash^* (q, \varepsilon)$. Since the language $L(A)$ is infinite, at least one word w in Σ^+ exists, such that $(q_0, w) \vdash^* (q_0, \varepsilon)$.

If the last character of the word w is a (that is, $w = ua$ for some u in Σ^*), we obtain

$$(q_0, uaa) \vdash^* (q_0, a) \vdash^* (p, \varepsilon).$$

However, since an accepting state is reachable from each state of the automaton A, it follows that a word in $L(A)$ exists, such that uaa is its prefix. But that clearly contradicts $L(A) = L^R$.

If the last character of the word w is b, it is possible to obtain a contradiction by a symmetrical argumentation. □

Theorem 2.2.19 The family $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ is not closed under homomorphism.

Proof. Let us consider a language $L = \{a,c\}^*$. It can be easily observed that L is in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. Now, let us consider a homomorphism $h : \{a,c\}^* \to \{a,b,c\}^*$ defined by

$$h(a) = ab,$$
$$h(c) = c.$$

We shall prove that the language $h(L) = \{ab,c\}^*$ is not in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$.

By contradiction. Let the language $h(L)$ be in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. Then, a strictly state-$\mathcal{A}$-equiloaded DFAε $A = (K,\Sigma,\delta,q_0,F)$ exists, such that $L(A) = h(L)$. Since $h(L)$ is infinite, the graphical representation of the automaton A is a directed multicycle through all states and the set of accepting states F is nonempty.

We shall prove that at least one word w in $L(A)$ exists, such that w is in $\{a,c\}^+$ and $\#_a(w) \geq 1$. This would lead to a contradiction, since there clearly is not any such word in $h(L)$.

The automaton A clearly has at least one transition on a, at least one transition on b, and at least one transition on c – otherwise, it could not accept the language $h(L)$. Moreover, if

$$(p,b) \vdash^* (q,\varepsilon)$$

for some p,q in K, then also

$$(p,a) \vdash^* (q,\varepsilon)$$

or

$$(p,c) \vdash^* (q,\varepsilon)$$

(since there is a word in $h(L)$, that does not contain any occurrence of b). But this, together with the obvious reachability of some transition on a, clearly implies that the automaton A accepts at least one word w in $\{a,c\}^+$, such that $\#_a(w) \geq 1$. □

Theorem 2.2.20 The family $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ is not closed under inverse homomorphism.

Proof. Let us consider a language $L = \{a\}$. Since the language L is finite, it belongs to the family $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. Moreover, let us consider a homomorphism $h : \{a,b\}^* \to \{a,b\}^*$ defined by

$$h(a) = a,$$
$$h(b) = \varepsilon.$$

We shall prove that the language $h^{-1}(L) = \{w \in \{a,b\}^* \mid \#_a(w) = 1\}$ is not in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$.

By contradiction. Let us suppose that $h^{-1}(L)$ is in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. Then a strictly state-$\mathcal{A}$-equiloaded DFA$\varepsilon$ $A = (K,\Sigma,\delta,q_0,F)$ exists, such that $L(A) = h^{-1}(L)$. Since the language $h^{-1}(L)$ is infinite, the graphical representation of the automaton A is a directed multicycle through all states, and thus, this graphical representation is strongly connected. Moreover, the automaton A has at least one accepting state.

The automaton a has to have at least one transition on a, since otherwise it could not accept the language $h^{-1}(L)$. However, the strong connectedness of the graphical representation of the automaton A implies that an accepting computation path γ of the automaton A exists, such that this transition on a is used at least twice in γ. Thus, a word w in $L(A)$ exists, such that $\#_a(w) \geq 2$, and that is an obvious contradiction. □

Closure Properties of the Family $\mathscr{L}_{\delta-SEQ-DFA}$

We have proved the closure properties of the family $\mathscr{L}_{\delta-SEQ-DFA}(\mathcal{A}) = \mathscr{L}_{\delta-SEQ-DFA}$ already in [25]. Thus, we shall omit the proof of the following theorem.

Theorem 2.2.21 The family $\mathscr{L}_{\delta-SEQ-DFA}$ is closed under intersection and not closed under concatenation, union, complementation, Kleene star, Kleene plus, reversal, homomorphism, and inverse homomorphism.

Closure Properties of the Family $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$

Now, we shall study the closure properties of the family $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. The closure properties of this family have not been studied yet, so we shall include the proofs as well.

Theorem 2.2.22 The family $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ is not closed under concatenation.

Proof. Let us consider languages $L_1 = \{a\}^*$ and $L_2 = \{b\}$. As can be easily seen, both L_1 and L_2 are in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. However, in the proof of Theorem 2.2.11, we have observed that their concatenation, i.e., the language

$$L_1 \cdot L_2 = \{a\}^* \cdot \{b\} = \{a^i b \mid i \in \mathbb{N}\},$$

is not in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. Thus, by applying Theorem 2.2.9, we may also conclude that $L_1 \cdot L_2$ is not in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. □

Theorem 2.2.23 The family $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ is not closed under union.

Proof. Again, let us consider languages $L_1 = \{a\}^*$ and $L_2 = \{b\}$, both in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. In the proof of Theorem 2.2.12, it have been shown that their union $L_1 \cup L_2$ is not in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. Thus, the language $L_1 \cup L_2$ is also not in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. □

Theorem 2.2.24 The family $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ is closed under intersection.

Proof. Let L_1, L_2 be languages in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. Then, by Theorem 2.2.9, they are also in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ and by Theorem 2.2.14, also their intersection $L_1 \cap L_2$ is in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$.

Now, if the intersection $L_1 \cap L_2$ is finite, then it also is in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$, and the theorem is proved. Thus, let us suppose that this intersection is infinite. Then, a DFAε $A = (K, \Sigma, \delta, q_0, F)$ exists, such that the graphical representation of the automaton A is a directed multicycle through all states.

Since both L_1 and L_2 are infinite languages in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$, for each n in $\mathbb{N}$ there is at most one word of length n in L_1, and at most one word of length n in L_2. Thus, there is at most one word of length n in $L_1 \cap L_2$. This implies that the graphical representation of the automaton A is in fact a directed cycle, i.e., the language $L_1 \cap L_2$ is in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. □

Theorem 2.2.25 The family $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ is not closed under complementation.

Proof. Let us consider a language $L = \{\varepsilon, a\}$ that is clearly in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. In the proof of Theorem 2.2.15, we have observed that the language

$$L^C = \{a^k \mid k \geq 2\}$$

is not in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$, and thus also not in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. □

Theorem 2.2.26 The family $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ is not closed under closure.

Proof. Let us consider a language $L = \{aa, bb\}$. This language is finite, and thus in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. However, in the proof of the Theorem 2.2.16, it was shown that the language $L^* = \{aa, bb\}^*$ is not in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$, and thus also not in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. □

Theorem 2.2.27 The family $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ is not closed under positive closure.

Proof. Exactly the same counterexample and an analogous argumentation can be used, as in the proof of Theorem 2.2.26 (see also the proof of Theorem 2.2.17). □

Theorem 2.2.28 The family $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ is not closed under reversal.

Proof. Let us consider a language $L = \{ab\}^*\{\varepsilon, a\}$. Clearly, L is in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. However, in the proof of Theorem 2.2.18, we have observed that its reversal, the language $L^R = \{\varepsilon, a\}\{ba\}^*$, is not in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$. Thus, the language L^R is also not in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. □

Theorem 2.2.29 The family $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ is closed under homomorphism.

Proof. Let L be a language in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ over an alphabet Σ, let $h : \Sigma^* \to \Gamma^*$ be a homomorphism. We shall prove that the language $h(L)$ is in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$.

Since L is in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$, a strictly transition-$\mathcal{A}$-equiloaded DFA$\varepsilon$ $A = (K, \Sigma, \delta, q_0, F)$ exists, such that $L(A) = L$. If the language L is finite, then also the language $h(L)$ is finite, and thus also in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. Let us therefore suppose that the language L is infinite. Then, the graphical representation of the automaton A is a directed cycle through all states.

We shall now construct a DFAε $A' = (K', \Gamma, \delta', q'_0, F')$, such that its graphical representation is a directed cycle through all states, and $L(A') = h(L)$. The idea of the following construction is to replace each transition of the automaton A with a *nonempty*[8] sequence of transitions, so that if the transition of A is on c in $\Sigma \cup \{\varepsilon\}$, then the word read after going through the corresponding sequence of transitions in A', is $h(c)$.

Formally, we shall construct the automaton A' as follows:

$$K' = K \cup \left(\bigcup_{\substack{q \in K \\ c \in \Sigma \\ \exists p \in K : \delta(q,c) = p \\ i \in \{2, \ldots, |h(c)|\}}} \{p_{q,c,i}\} \right),$$

$q'_0 = q_0$, $F' = F$. Now, if $\delta(q, c) = p$ for some states p, q in K and some c in $\Sigma \cup \{\varepsilon\}$, let us denote $p_{q,c,1} := q$. Then,

$$\forall p_{q,c,i} \in K' : \delta'(p_{q,c,i}, h(c)[i]) = \begin{cases} p & \text{if } \delta(q,c) = p \ \wedge \ i = \max\{1, |h(c)|\} \\ p_{q,c,(i+1)} & \text{if } i < |h(c)| \end{cases} .$$

It is clear that $L(A') = h(L)$ and that the automaton A' is strictly transition-$\mathcal{A}$-equiloaded. Thus, the language $h(L)$ is in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. □

Theorem 2.2.30 The family $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ is not closed under inverse homomorphism.

Proof. Let us consider a language $L = \{a\}$ and a homomorphism $h : \{a, b\}^* \to \{a, b\}^*$ defined by

$$h(a) = a,$$
$$h(b) = \varepsilon.$$

Since the language L is finite, it belongs to the family $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. However, in the proof of Theorem 2.2.20, we have observed that the language

$$h^{-1}(L) = \{w \in \{a, b\}^* \mid \#_a(w) = 1\}$$

is not in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$, and thus also not in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. □

[8] The nonemptiness of that sequence of transitions is of crucial importance here. This is also the reason, why the family $\mathscr{L}_{\delta-SEQ-DFA}$ is not closed under homomorphism: without the use of ε-transitions, there is not any such nonempty sequence for the case $h(c) = \varepsilon$.

2.3 $\mathcal{S}$-Equiloadedness

In this section, we shall study $\mathcal{S}$-equiloaded DFA and DFAε. However, in most results presented in this section, we shall not be concerned with $\mathcal{S}$-equiloadedness for general $\mathcal{S}$, but with several special cases of $\mathcal{S}$. The most important choices of $\mathcal{S}$ are $\mathcal{C}_=, \mathcal{C}_\leq, \mathcal{A}_=$, and $\mathcal{A}_\leq$.

Some of the families of $\mathcal{S}$-equiloaded automata and languages have already been studied (although the terminology and definitions have been slightly different, since the concept of $\mathcal{S}$-equiloadedness is introduced in this report as a generalization of the older definitions).

In [26] and [27], state-$\mathcal{A}_=$-equiloaded deterministic finite automata without ε-transitions have been studied. However, the definition used in [26] and [27] is conceptually different from the definition used in this report and, up to now, the equivalence of both definitions has been an open problem. This open problem is solved in Subsection 2.3.1. Moreover, in Subsection 2.3.1, analogous equivalence theorems are proved also for $\mathcal{S} = \mathcal{C}_=$, and for deterministic finite automata with ε-transitions.

In [25], we have studied transition-$\mathcal{A}_=$-equiloaded deterministic finite automata without ε-transitions. We have used the definition from [25] as a basis for the more general definition of $\mathcal{S}$-equiloadedness used in this report, and can therefore be viewed as its special case. Thus, there is no need to prove the equivalence. However, in [25], we have proved the equivalence of the alternative definition of transition-$\mathcal{A}_=$-equiloadedness analogous to the definitions used in [26] and [27] for state-equiloadedness and, in Subsection 2.3.1 of this report, we shall prove that these alternative definitions are equivalent to transition-$\mathcal{S}$-equiloadedness also for several other choices of $\mathcal{S}$, and for DFAε.

In addition to alternative definitions, we shall also study some other aspects of $\mathcal{S}$-equiloadedness in this section. In [25], we have presented the characterization of weakly transition-$\mathcal{A}_=$-equiloaded DFA. In Subsection 2.3.5, we shall generalize this result to several other choices of $\mathcal{S}$ and to DFAε. Moreover, we shall prove also the characterization of weakly state-$\mathcal{C}_=$-equiloaded DFA and DFAε. Subsequently, we shall be interested in the relations between corresponding families of languages. Finally, in Subsection 2.3.7, we shall examine the closure properties of families of languages corresponding to $\mathcal{S}$-equiloaded DFA.

2.3.1 Alternative Definitions of $\mathcal{S}$-Equiloadedness for $\mathcal{S} = \mathcal{C}_=$ and $\mathcal{S} = \mathcal{A}_=$

In this subsection, we shall prove a theorem providing alternative definitions of state-$\mathcal{S}$-equiloadedness and transition-$\mathcal{S}$-equiloadedness for two most important choices of $\mathcal{S}$ – for $\mathcal{C}_=$, and $\mathcal{A}_=$. The basic definitions of the concept of $\mathcal{S}$-equiloadedness via the measure of $\mathcal{S}$-equiloadedness, as presented in Section 1.5, are a generalization of the definition of transition-$\mathcal{A}_=$-equiloaded DFA that we have used in [25]. The alternative definitions of $\mathcal{S}$-equiloadedness, provided by Theorem 2.3.4 presented in this subsection, are based on the definitions of state-$\mathcal{A}_=$-equiloaded DFA used in [26] and [27].

However, before we present these alternative definitions, we shall prove three lemmas that we shall use in the proof of Theorem 2.3.4. First, in Lemma 2.3.1, we shall observe that the asymptotic growth of the quantity representing the number of computation paths of length n in a strongly connected automaton is determined by the Perron-Frobenius eigenvalue of the transition matrix.

Lemma 2.3.1 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with strongly connected graphical representation, and Δ be the transition matrix of the automaton A. Let ρ be the Perron-Frobenius eigenvalue of the transition matrix Δ. Then

$$F_0(n) = |\mathrm{Comp}(A, n)| = \Theta(\rho^n).$$

Proof. Since the graphical representation of the automaton A is strongly connected, the transition matrix Δ is either an irreducible matrix, or the 1×1 null matrix. For the case of the 1×1 null matrix, the statement of the lemma is clearly true for $\rho = 0$ (i.e., the 1×1 null matrix can be assumed to have a zero Perron-Frobenius eigenvalue). In the rest of the proof, we shall therefore assume that the matrix Δ is irreducible.

By the Perron-Frobenius theorem, the transition matrix Δ has the Perron-Frobenius eigenvalue, i.e., the statement of the lemma makes sense. Moreover, the eigenvalues of Δ are complex numbers

$$\rho, \rho \cdot e^{2\pi i/p}, \ldots, \rho \cdot e^{2\pi i(p-1)/p}, \lambda_1, \ldots, \lambda_k,$$

where p in $\mathbb{N}^+$ is a positive integer, k in $\mathbb{N}$ is a nonnegative integer and

$$|\lambda_j| < \rho,$$

for $j = 1, \ldots, k$. Moreover, all eigenvalues of the absolute value ρ are simple, i.e., of algebraic multiplicity 1. Let us denote the algebraic multiplicities of $\lambda_1, \ldots, \lambda_k$ by $\alpha_1, \ldots, \alpha_k$.

Since the graphical representation of the automaton A is strongly connected with at least one transition (a strongly connected digraph does not have any edge only if its adjacency matrix is the 1×1 null matrix), it is clear that $F_0(n)$ is nonzero for all n. Thus, by the results obtained in Section 2.1, the transition matrix Δ has at least one nonzero eigenvalue (otherwise, every possible solution for $F_0(n)$ would be nonzero only for finite number of n) and, for n greater than some n_0 in $\mathbb{N}$, the property

$$\begin{aligned} F_0(n) &= \sum_{j=0}^{p-1} c_j \cdot \rho^n \cdot e^{2\pi inj/p} + \sum_{j=1}^{k} \sum_{h=0}^{\alpha_j - 1} c_{j,h} \cdot n^h \cdot \lambda_j^n = \\ &= \left(\sum_{j=0}^{p-1} c_j \cdot e^{2\pi inj/p} \right) \cdot \rho^n + \sum_{j=1}^{k} \sum_{h=0}^{\alpha_j - 1} c_{j,h} \cdot n^h \cdot \lambda_j^n \end{aligned} \tag{2.6}$$

holds (terms corresponding to zero eigenvalues have been omitted, since they have effect on the value of $F_0(n)$ only for the finite number of n) for some complex constants c_j, $j = 0, \ldots, p-1$ and $c_{j,h}$, $j = 1, \ldots, k$, $h = 0, \ldots, \alpha_j - 1$.

The function (of the variable n)

$$\sum_{j=0}^{p-1} c_j \cdot e^{2\pi inj/p}, \tag{2.7}$$

arising in (2.6), is clearly periodic with period p. We shall prove that the value of the function (2.7) is nonzero for all n in $\mathbb{N}$.

By the results obtained in Section 2.1,

$$\mathbf{F}_n = \Delta^n \cdot \mathbf{F}_0,$$

with

$$\mathbf{F}_n = (F_0(n), F_1(n), \ldots, F_{m-1}(n))^T$$

and

$$\mathbf{F}_0 = (\underbrace{1, 1, \ldots, 1}_{m})^T,$$

where $m = |K|$. Let us denote by E^m the *standard* $(m-1)$*-simplex*, i.e., the set

$$E^m = \left\{ (x_1, \ldots, x_m) \in \mathbb{R}^m \;\middle|\; x_1 \geq 0, \ldots, x_m \geq 0, \sum_{i=1}^{m} x_i = 1 \right\}.$$

Now, let $\mathbf{x}_\rho$ in E^m be the Perron-Frobenius eigenvector of the matrix Δ, corresponding to the Perron-Frobenius eigenvalue ρ. Since $\mathbf{x}_\rho$ is in E^m,

$$\mathbf{F}_0 \geq \mathbf{x}_\rho.$$

Thus,

$$\mathbf{F}_n = \Delta^n \cdot \mathbf{F}_0 \geq \Delta^n \cdot \mathbf{x}_\rho = \rho^n \cdot \mathbf{x}_\rho.$$

Since the Perron-Frobenius eigenvector $\mathbf{x}_\rho$ is always positive, we may conclude that a positive real constant Q in $\mathbb{R}^+$ exists, such that

$$F_0(n) \geq Q \cdot \rho^n. \tag{2.8}$$

Now, for the purpose of contradiction, let us suppose that (2.7) is zero for some n in $\mathbb{N}$. Then, the periodicity implies that (2.7) is zero for infinitely many n in $\mathbb{N}$. However, this clearly contradicts (2.8). Thus, the value of the function (2.7) is nonzero for all n in $\mathbb{N}$.

Moreover, the function (2.7) has to be real. Otherwise, for some j in $\{0, \ldots, p-1\}$, the function would have to attain a nonreal complex value $z = a + bi$, $b \neq 0$, for all $n \cdot p + j$, n in $\mathbb{N}$. However, for n greater than some n_0 in $\mathbb{N}$, clearly

$$\left| \sum_{j=1}^{k} \sum_{h=0}^{\alpha_j - 1} c_{j,h} \cdot n^h \cdot \lambda_j^n \right| < b \cdot \rho^n$$

and thus, the function $F_0(n)$ would have to attain nonreal complex values for n greater than n_0. However, $F_0(n)$ is clearly a real function. Thus, the function (2.7) has to be real as well.

Further, we shall prove that the function (2.7) is not only real, but in fact positive. We have already proved that the value of the function (2.7) at given n is real and nonzero. For the purpose of contradiction, let us suppose that it attains a negative value for some n. Then, by the periodicity of the function (2.7), it follows that (2.7) attains a negative value for infinitely many n. However, by applying a similar argument as above, it can be easily seen that this implies that also the function $F_0(n)$ attains at least one negative value. But this is a contradiction, since the function $F_0(n)$ is clearly nonnegative.

Thus, we have proved that (2.7) is a positive function periodic with period p. This implies that it has to be bounded from below and from above by a positive constant. This clearly implies

$$F_0(n) = |\mathrm{Comp}(A, n)| = \Theta(\rho^n).$$

Thus, the lemma is proved. □

Next, in Lemma 2.3.2, we shall prove that a similar property as in Lemma 2.3.1 holds also for the number of all *accepting* computation paths of a given length. The only difference is that in this case, lengths n, such that there is not any accepting computation of length n, are a serious problem. However, if we omit these lengths from our consideration, the property holds.

Lemma 2.3.2 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with strongly connected graphical representation, and Δ be the transition matrix of the automaton A. Let ρ be the Perron-Frobenius eigenvalue of the transition matrix Δ. Let F be nonempty and $\{n_k\}_{k=0}^{\infty}$ be the increasing sequence of all nonnegative integers n_k, such that at least one accepting computation path of length n_k exists in the automaton A. Then

$$f_0(n_k) = |\mathrm{Acc}(A, n_k)| = \Theta(\rho^{n_k}).$$

Proof. The statement of the lemma is obviously true in the case when Δ is the 1×1 null matrix. Thus, let us suppose that Δ is irreducible. Then $F_0(n)$ is nonzero for all n in $\mathbb{N}$ and thus, by Lemma 2.3.1, the Perron-Frobenius eigenvalue ρ of the matrix Δ is nonzero (this can be of course easily proved also without the use of Lemma 2.3.1).

Let us denote by d the greatest common divisor of lengths of the closed walks in the graphical representation of the automaton A, with the beginning and end in the vertex corresponding to the initial state q_0. Let q in F be an accepting state, and r in $\mathbb{N}$ be a nonnegative integer, such that

$$(q_0, w) \vdash^r (q, \varepsilon)$$

for some w in Σ^*.

Let S be a set of nonnegative integers s, such that a word w in Σ^* exists, such that

$$(q_0, w) \vdash^s (q_0, \varepsilon).$$

Clearly, the greatest common divisor of elements of S is d, and the set S is closed under addition. Thus, by Lemma A.5.3, a nonnegative integer n_0 in $\mathbb{N}$ exists, such that for all n in $\mathbb{N}$, $n \geq n_0$, such that d divides n, n is in S. Let n_1 be the smallest nonnegative integer greater than or equal to n_0, such that d divides n_1.

Let k be in $\mathbb{N}$. By Lemma 2.3.1,

$$\begin{aligned} F_0(n_k - (|K|-1) - n_1 - r) &= |\mathrm{Comp}(A, n_k - (|K|-1) - n_1 - r)| = \Theta(\rho^{n_k-(|K|-1)-n_1-r}) = \\ &= \Theta(\rho^{n_k}). \end{aligned}$$

Now, since the graphical representation of the automaton A is strongly connected, each state is reachable from each other state in at most $|K|-1$ steps. Thus, every computation path of length $n_k - (|K|-1) - n_1 - r$ can be prolonged to a computation path of length at most $n_k - n_1 - r$, ending in state q_0. Moreover, if two computation paths are distinct, then every two computation paths obtained by prolonging these two computation paths are distinct as well. Thus, if we denote by $\varphi(n_k - n_1 - r)$ the number of maximal[9] computation paths of length at most $n_k - n_1 - r$ ending in q_0, we have

$$\varphi(n_k - n_1 - r) \geq F_0(n_k - (|K|-1) - n_1 - r) = \Theta(\rho^{n_k}),$$

i.e.,

$$\varphi(n_k - n_1 - r) = \Omega(\rho^{n_k}).$$

Clearly, all of these $\varphi(n_k - n_1 - r)$ computation paths are of length divisible by d. Thus, by the structure of the set S, all of these $\varphi(n_k - n_1 - r)$ can be prolonged to a computation path of length $n_k - r$ ending in q_0. Since the computation paths involved are maximal, the prolonged computation paths are all distinct. Thus, if we denote by $\psi(n_k - r)$ the number of computation paths of length exactly $n_k - r$ ending in q_0, we have

$$\psi(n_k - r) \geq \varphi(n_k - n_1 - r) = \Omega(\rho^{n_k}),$$

i.e.,

$$\psi(n_k - r) = \Omega(\rho^{n_k}).$$

However, since the accepting state q is reachable from q_0 in r steps, this implies

$$f_0(n_k) = |\mathrm{Acc}(A, n_k)| \geq \psi(n_k - r) = \Omega(\rho^{n_k}),$$

i.e.,

$$f_0(n_k) = |\mathrm{Acc}(A, n_k)| = \Omega(\rho^{n_k}).$$

However, on the other hand we have

$$f_0(n_k) = |\mathrm{Acc}(A, n_k)| \leq |\mathrm{Comp}(A, n_k)| = F_0(n_k) = \Theta(\rho^{n_k}),$$

i.e.,

$$f_0(n_k) = |\mathrm{Acc}(A, n_k)| = O(\rho^{n_k}).$$

Thus,

$$f_0(n_k) = |\mathrm{Acc}(A, n_k)| = \Theta(\rho^{n_k})$$

and the lemma is proved. □

Finally, in Lemma 2.3.3, we shall state one easy observable yet useful property that we shall use later in our study.

[9] In the sense that a computation path cannot be prolonged to another computation path satisfying the property that its length is at most $n_k - n_1 - r$ and it ends in q_0.

Lemma 2.3.3 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε, and Δ be the transition matrix of the automaton A. The transition matrix Δ has a nonzero eigenvalue if and only if the graphical representation of the automaton A contains at least one directed cycle.

Proof. Without loss of generality, let us suppose that the matrix Δ is in the normal form of a reducible matrix. The spectrum of the matrix Δ is the union of spectra of its diagonal blocks (corresponding to strongly connected components of the graphical representation of the automaton A) that are either irreducible or 1×1 null matrices.

If the graphical representation of the automaton A contains at least one directed cycle, then there is at least one strongly connected component with at least one directed cycle. Then, the number of computation paths in this strongly connected component (with arbitrary initial state) of length n has to be nonzero for all n in $\mathbb{N}$. Thus, by Lemma 2.3.1, the corresponding diagonal block has to have at least one nonzero eigenvalue (the Perron-Frobenius eigenvalue of that block).

If the graphical representation does not contain any directed cycle, then each state forms one strongly connected component. The number of computation paths in all strongly connected components is thus 1 for the computation paths of length 0 and 0 otherwise. Thus, by Lemma 2.3.1, the eigenvalue of each diagonal 1×1 block is 0 (that is, all diagonal blocks are null), and the matrix Δ does not have any nonzero eigenvalue. □

Now we are prepared to prove the key result of this subsection, providing an alternative definition of $\mathcal{S}$-equiloadedness, for $\mathcal{S} = \mathcal{C}_=$ and $\mathcal{S} = \mathcal{A}_=$. The following theorem implies that a definition of DFA with balanced use of states in accepting computations, used in [26] and [27], is equivalent to our definition of state-$\mathcal{A}_=$-equiloadedness, and that a similar property holds also for some other types of $\mathcal{S}$-equiloadedness defined in this report.

Theorem 2.3.4 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε. Let $\mathcal{S}$ be a function in $\{\mathcal{C}_=, \mathcal{A}_=\}$.

a) A is state-$\mathcal{S}$-equiloaded if and only if a real constant η in $\mathbb{R}$ exists, such that, for all pairs of states p, q in K and for all n in $\mathbb{N}$, the property

$$|\#[p, \mathcal{S}(A, n)] - \#[q, \mathcal{S}(A, n)]| \leq \eta \cdot |\mathcal{S}(A, n)|$$

holds.

b) A is transition-$\mathcal{S}$-equiloaded if and only if a real constant η in $\mathbb{R}$ exists, such that, for all pairs of transitions e, f in D and for all n in $\mathbb{N}$, the property

$$|\#[e, \mathcal{S}(A, n)] - \#[f, \mathcal{S}(A, n)]| \leq \eta \cdot |\mathcal{S}(A, n)|$$

holds.

Proof. We shall prove only the statement for transitions, the proof of the statement for states is analogous.

We shall start with the proof of the easier right-to-left implication. Clearly, since every computation path of length n consists of n transition uses, and since both $\mathcal{C}_=(A, n) = \mathrm{Comp}(A, n)$ and $\mathcal{A}_=(A, n) = \mathrm{Acc}(A, n)$ contain only computation paths of length n, we have

$$\sum_{e \in D} \#[e, \mathcal{S}(A, n)] = n \cdot |\mathcal{S}(A, n)|.$$

Thus,

$$\frac{n}{|D|} \cdot |\mathcal{S}(A, n)| \leq \max_{e \in D} \#[e, \mathcal{S}(A, n)] \leq n \cdot |\mathcal{S}(A, n)|,$$

i.e.,

$$\max_{e \in D} \#[e, \mathcal{S}(A, n)] = g(n) |\mathcal{S}(A, n)|,$$

where $g(n)$ is $\Theta(n)$. Now, if the inequality

$$|\#[e, \mathcal{S}(A, n)] - \#[f, \mathcal{S}(A, n)]| \leq \eta \cdot |\mathcal{S}(A, n)|$$

holds for all e, f in D and for all n in $\mathbb{N}$, then the number of uses of each transition e in D can be written as

$$\#[e, \mathcal{S}(A,n)] = (g(n) + r_e(n))|\mathcal{S}(A,n)|,$$

where $r_e : \mathbb{N} \to \mathbb{R}$ is a function, such that $|r_e(n)| = O(1)$.

Now, if $|\mathcal{S}(A,n)|$ is zero for all $n \geq n_0$ for some n_0 in $\mathbb{N}$, then, applying Lemma 1.5.3,

$$B_A(\mathcal{S}) = \min_{(e,f)\in D^2} \liminf_{n\to\infty} \frac{(g(n)+r_e(n))|\mathcal{S}(A,n)|+1}{(g(n)+r_f(n))|\mathcal{S}(A,n)|+1} = \liminf_{n\to\infty} \frac{1}{1} = 1.$$

Thus, in this case the implication holds.

Now, let us suppose that $|\mathcal{S}(A,n_k)| > 0$ for infinitely many nonnegative integers n_k, $k = 0,1,2,\ldots$. Then,

$$\begin{aligned} B_A(\mathcal{S}) &= \min_{(e,f)\in D^2} \liminf_{n\to\infty} \frac{(g(n)+r_e(n))|\mathcal{S}(A,n)|+1}{(g(n)+r_f(n))|\mathcal{S}(A,n)|+1} = \\ &= \min_{(e,f)\in D^2} \liminf_{k\to\infty} \frac{(g(n_k)+r_e(n_k))|\mathcal{S}(A,n_k)|}{(g(n_k)+r_f(n_k))|\mathcal{S}(A,n_k)|} = \\ &= \min_{(e,f)\in D^2} \liminf_{k\to\infty} \frac{g(n_k)+r_e(n_k)}{g(n_k)+r_f(n_k)} = \min_{(e,f)\in D^2} \liminf_{k\to\infty} \frac{\frac{g(n_k)}{n_k}+\frac{r_e(n_k)}{n_k}}{\frac{g(n_k)}{n_k}+\frac{r_f(n_k)}{n_k}} = 1, \end{aligned}$$

since both $\frac{r_e(n_k)}{n_k}$ and $\frac{r_f(n_k)}{n_k}$ tend to 0 as k goes to infinity and $\frac{g(n_k)}{n_k}$ is bounded from below (and also from above) by a positive constant. Thus, the first implication is proved.

Now, let us prove the remaining left-to-right implication. Let $\mathcal{S}$ be in $\{\mathcal{C}_=, \mathcal{A}_=\}$. Let us suppose that the automaton A is transition-$\mathcal{S}$-equiloaded. We shall show that the alternative definition applies to A.

If the graphical representation of the automaton A does not contain any directed cycle, the implication clearly holds. Let us therefore suppose that there is a directed cycle in the graphical representation of the automaton A.

Let $f : \mathbb{N} \to \mathbb{N}$ denote the function $F_0(n) = |\mathrm{Comp}(A,n)|$ for the case of $\mathcal{S} = \mathcal{C}_=$, and the function $f_0(n) = |\mathrm{Acc}(A,n)|$ for the case of $\mathcal{S} = \mathcal{A}_=$.

Let m be a number of states of the automaton A. For the transition matrix Δ of the automaton A, an $m \times m$ permutation matrix P exists, such that $P \cdot \Delta \cdot P^{-1}$ is in the normal form of a reducible matrix. That is, for some μ in $\mathbb{N}$,

$$P \cdot \Delta \cdot P^{-1} = \begin{pmatrix} \Delta_{1,1} & \Delta_{1,2} & \ldots & \Delta_{1,\mu} \\ \mathbf{0} & \Delta_{2,2} & \ldots & \Delta_{2,\mu} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \Delta_{\mu,\mu} \end{pmatrix},$$

where $\Delta_{j,j}$ are square blocks for $j = 1,\ldots,\mu$ that are either irreducible or 1×1 null matrices. Since the matrices Δ and $P \cdot \Delta \cdot P^{-1}$ are similar, they have the same spectrum. Moreover, since the matrix $P \cdot \Delta \cdot P^{-1}$ is an upper triangular block matrix, its spectrum is the union of the spectra of the blocks $\Delta_{j,j}$, for $j = 1,\ldots,\mu$. Thus, also the spectrum of the transition matrix Δ is the union of the spectra of the blocks $\Delta_{j,j}$, for $j = 1,\ldots,\mu$.

Let us denote the Perron-Frobenius eigenvalue of the irreducible (or null) block $\Delta_{j,j}$ by ρ'_j (if $\Delta_{j,j}$ is a null block, then $\rho'_j = 0$), for $j = 1,\ldots,\mu$. Moreover, let p_j, for $j = 1,\ldots,\mu$, denote the greatest natural number, such that

$$\rho'_j, \rho'_j \cdot e^{2\pi i/p_j}, \ldots, \rho'_j \cdot e^{2\pi i(p_j-1)/p_j}$$

are eigenvalues of $\Delta_{j,j}$. By the Perron-Frobenius theorem, all these are simple eigenvalues of $\Delta_{j,j}$ (although *not necessarily simple* eigenvalues of Δ). Let us denote the pairwise distinct nonzero eigenvalues ρ, such that $\rho = \rho'_j$ for some $j = 1, \ldots, \mu$, by

$$\rho_1, \ldots, \rho_r,$$

where r is the number of such pairwise distinct nonzero eigenvalues. Since the graphical representation of the automaton A contains a directed cycle, it follows from Lemma 2.3.3 that $r \geq 1$.

Now, let us consider the spectrum of the transition matrix Δ. Let S be a set of all pairs of nonnegative integers (s_1, s_2) in $\mathbb{N}^2$, such that $s_1 < s_2$, s_1, s_2 are coprime and $\rho_j \cdot e^{2\pi i s_1/s_2}$ is an eigenvalue of Δ, for some j in $\{1, \ldots, r\}$. The set S is clearly finite. For given j in $\{1, \ldots, r\}$ and (s_1, s_2) in S, let us denote by β_{j,s_1,s_2} the algebraic multiplicity of the eigenvalue $\rho_j \cdot e^{2\pi i s_1/s_2}$ (more precisely, $\rho_j \cdot e^{2\pi i s_1/s_2}$ need not to be an eigenvalue, since the pair (s_1, s_2) in S may not correspond to j – in that case we define $\beta_{j,s_1,s_2} = 0$). Moreover, let us denote the (pairwise distinct) nonzero eigenvalues that are not of this form by $\lambda_1, \ldots, \lambda_k$ and their algebraic multiplicities by $\alpha_1, \ldots, \alpha_k$, where k in $\mathbb{N}$ is the number of pairwise distinct eigenvalues of Δ that are not of the above defined form. By the results obtained in Section 2.1, for n greater than some n_0 in $\mathbb{N}$ (we shall assume this in the rest of the proof),

$$f(n) = \sum_{j=1}^{r} \sum_{(s_1,s_2)\in S} \sum_{h=0}^{\beta_{j,s_1,s_2}-1} c_{j,s_1,s_2,h} \cdot n^h \cdot \rho_j^n \cdot e^{2\pi i n s_1/s_2} + \sum_{j=1}^{k} \sum_{h=0}^{\alpha_j-1} c_{j,h} \cdot n^h \cdot \lambda_j^n,$$

where

$$c_{j,s_1,s_2,h}, \qquad j = 1, \ldots, r, (s_1, s_2) \in S, h = 0, \ldots, \beta_{j,s_1,s_2} - 1$$

and

$$c_{j,h}, \qquad j = 1, \ldots, k, h = 0, \ldots, \alpha_j - 1$$

are constants. Clearly, we may replace each β_{j,s_1,s_2} by the number

$$\beta = \max_{\substack{j=1,\ldots,r \\ (s_1,s_2)\in S}} \beta_{j,s_1,s_2}$$

by setting the newly introduced constants $c_{j,s_1,s_2,h}$ to 0. Thus, we obtain

$$f(n) = \sum_{j=1}^{r} \sum_{h=0}^{\beta-1} n^h \cdot \left(\sum_{(s_1,s_2)\in S} c_{j,s_1,s_2,h} \cdot e^{2\pi i n s_1/s_2} \right) \cdot \rho_j^n + \sum_{j=1}^{k} \sum_{h=0}^{\alpha_j-1} c_{j,h} \cdot n^h \cdot \lambda_j^n. \tag{2.9}$$

Next, since $e^{2\pi i N} = 1$ for all nonnegative integers N in $\mathbb{N}$, the function

$$\sum_{(s_1,s_2)\in S} c_{j,s_1,s_2,h} \cdot e^{2\pi i n s_1/s_2} \tag{2.10}$$

(of variable n) is clearly periodic with period s, where s is the least common multiple of all nonnegative integers s_2 in $\mathbb{N}$, such that (s_1, s_2) is in S for some s_1 in $\mathbb{N}$ (the set S is finite, so the least common multiple is well-defined).

Now, for $a = 0, \ldots, s-1$, let us denote by $\varphi_a : \mathbb{N} \to \mathbb{N}$ the function defined by

$$\varphi_a(n) = f(n \cdot s + a)$$

for n in $\mathbb{N}$. From (2.9) and from the periodicity of the function (2.10), it follows that

$$\varphi_a(n) = \sum_{j=1}^{r} \sum_{h=0}^{\beta-1} (n \cdot s + a)^h \cdot C_{j,h} \cdot \rho_j^{ns+a} + \sum_{j=1}^{k} \sum_{h=0}^{\alpha_j-1} c_{j,h} \cdot (n \cdot s + a)^h \cdot \lambda_j^{ns+a} \tag{2.11}$$

for $a = 0, \ldots, s-1$, where $C_{j,h}$ is a complex constant for $j = 1, \ldots, r$ and $h = 0, \ldots, \beta - 1$. Moreover, in a similar way as in the proof of Lemma 2.3.1, it is possible to prove that the constants $C_{j,h}$ are in fact real.

Let a in $\{0, \ldots, s-1\}$ be fixed. We shall show that if the greatest Perron-Frobenius eigenvalue, for which some nonzero coefficient $C_{j,h}$ exists in (2.11), is ρ_J for some J in $\{1, \ldots, r\}$, then each eigenvalue λ_j (with j in $\{1, \ldots, k\}$) with some nonzero coefficient $c_{j,l}$ (with l in $\{0, \ldots, \alpha_j - 1\}$) in (2.11) is of absolute value strictly smaller then ρ_J.

We shall first prove this for the case $f(n) = F_0(n) = |\mathrm{Comp}(A, n)|$. For the purpose of contradiction, let us suppose that the converse is true, i.e., that an eigenvalue λ_j of absolute value greater than or equal to ρ_J exists, such that $c_{j,l}$ is nonzero in (2.11) for some l. This implies that a computation path γ of the automaton A exists, such that γ visits some state q belonging to the strongly connected component of the graphical representation, to which the eigenvalue λ_j corresponds (if there are more such components, then it visits at least one of them). Otherwise, the function $\varphi_a(n)$ would be the same as the function $\varphi_a(n)$ corresponding to the automaton with this strongly connected component deleted. This argument can be repeated while the reduced automaton has λ_j as an eigenvalue. At the end of this process, we obtain an automaton with the same $\varphi_a(n)$, but without λ_j as an eigenvalue. However, by linear independence of involved functions, this implies that all coefficients $c_{j,l}$, $l = 0, \ldots, \alpha_j - 1$ are zero, i.e., a contradiction.

This strongly connected component has a Perron-Frobenius eigenvalue ρ_b for some b taken from $\{1, \ldots, r\}$. Thus, clearly $\rho_b > |\lambda_j| \geq \rho_J$. Now, if we construct an automaton B from this strongly connected component by choosing q to be its initial state (the set of accepting states may be arbitrary), and if we denote by $\psi(n)$ the number of computation paths of length n in the automaton B, by Lemma 2.3.1 we have $\psi(n) = \Theta(\rho_b^n)$. Thus,

$$\varphi_a(n) = f(n \cdot s + a) = F_0(n \cdot s + a) \geq \psi(n \cdot s + a - |\gamma|) = \Omega\left(\rho_b^{ns+a-|\gamma|}\right),$$

since the computation path γ may continue by all of $\psi(n)$ computation paths of the automaton B. However, this is an obvious contradiction.

Now, let us prove the claim for the case $f(n) = f_0(n) = |\mathrm{Acc}(A, n)|$. Let us suppose that the converse is true. By a similar argument as in the previous case, a computation path γ of the automaton A exists, such that:

(i) The computation path γ is accepting.

(ii) The computation path γ is of length $|\gamma| = n_\gamma \cdot s + a$ for some n_γ in $\mathbb{N}$.

(iii) The computation path γ visits some state q of the strongly connected component, to which the eigenvalue λ_j corresponds. Let us denote by $B = (K', \Sigma, \delta', q_0', F')$ the automaton constructed from this component by setting $q_0' = q$ and $F' = \{q\}$.

Let us denote by d the greatest common divisor of lengths of closed walks in the graphical representation of the automaton B, beginning and ending in q. The set S of lengths of such closed walks is clearly closed under addition. Thus, by Lemma A.5.3 in Subsection A.5 of Appendix A, a positive integer n_0 in $\mathbb{N}$ exists, such that S contains all multiples of d greater than or equal to n_0. There are clearly infinitely many multiples of s in S. Let $\{n_k\}_{k=0}^{\infty}$ be the increasing sequence of all multiples of s in S. Since q is accepting, there is at least one accepting computation path of length n_k in B, for all k in $\mathbb{N}$. Let us denote by $\psi(n_k)$ the number of accepting computation paths of the automaton B of length n_k. By Lemma 2.3.2,

$$\psi(n_k) = |\mathrm{Acc}(B, n_k)| = \Theta(\rho_b^{n_k}),$$

where $\rho_b > |\lambda_j| \geq \rho_J$. Thus, if $n_k = m_k \cdot s$ for all k in $\mathbb{N}$, we have

$$\varphi_a(m_k + n_\gamma) = f((m_k + n_\gamma) \cdot s + a) = f_0((m_k + n_\gamma) \cdot s + a) \geq \psi(m_k \cdot s) = \psi(n_k) = \Theta(\rho_b^{n_k}).$$

This is an obvious contradiction.

Thus, we may conclude that

$$\varphi_a(n) = \Theta\left((n \cdot s + a)^H \cdot \rho_J^{ns+a}\right), \tag{2.12}$$

where H is a greatest nonnegative integer, such that $c_{J,H}$ is nonzero (by our assumptions, at least one such nonnegative integer exists).

Now, we can finally prove the left-to-right implication from the statement of the theorem. Let $g^{(1)} : \mathbb{N} \to \mathbb{N}$ denote, depending on the case for which the implication is being proved, one of the basic quantities $T_i^e(n) = \#[e, \mathrm{Comp}(A, n)]$, and $t_i^e(n) = \#[e, \mathrm{Acc}(A, n)]$. Let $g^{(2)} : \mathbb{N} \to \mathbb{N}$ denote one of the functions $T_i^f(n) = \#[f, \mathrm{Comp}(A, n)]$, and $t_i^f(n) = \#[f, \mathrm{Acc}(A, n)]$ (depending on the case for which the implication is being proved).

Moreover, let $\varphi_a^{(1)} : \mathbb{N} \to \mathbb{N}$ denote the function

$$\varphi_a^{(1)}(n) = g^{(1)}(n \cdot s + a)$$

for $a = 0, \ldots, s-1$, and $\varphi_a^{(2)} : \mathbb{N} \to \mathbb{N}$ denote the function

$$\varphi_a^{(2)}(n) = g^{(2)}(n \cdot s + a)$$

for $a = 0, \ldots, s-1$. Let a in $\{0, \ldots, s-1\}$ be fixed. By the results obtained in Section 2.1, and by the same argumentation as for the case of functions $f(n)$ and $\varphi_a(n)$, we obtain

$$\varphi_a^{(1)}(n) = \sum_{j=1}^{r} \sum_{h=0}^{2\beta-1} (n \cdot s + a)^h \cdot D_{j,h}^{(1)} \cdot \rho_j^{ns+a} + \sum_{j=1}^{k} \sum_{h=0}^{2\alpha_j-1} d_{j,h}^{(1)} \cdot (n \cdot s + a)^h \cdot \lambda_j^{ns+a}, \tag{2.13}$$

$$\varphi_a^{(2)}(n) = \sum_{j=1}^{r} \sum_{h=0}^{2\beta-1} (n \cdot s + a)^h \cdot D_{j,h}^{(2)} \cdot \rho_j^{ns+a} + \sum_{j=1}^{k} \sum_{h=0}^{2\alpha_j-1} d_{j,h}^{(2)} \cdot (n \cdot s + a)^h \cdot \lambda_j^{ns+a}, \tag{2.14}$$

where $D_{j,h}^{(1)}, D_{j,h}^{(2)}$, $j = 1, \ldots, r$, $h = 0, \ldots, 2\beta - 1$ are real constants, and $d_{j,h}^{(1)}, d_{j,h}^{(2)}$, $j = 1, \ldots, k$, $h = 0, \ldots, 2\alpha_j - 1$ are complex constants.

We assume that the automaton A is transition-$\mathcal{S}$-equiloaded. Moreover, for both $\mathcal{S}$ to which the statement of the theorem applies,

$$\sum_{y \in D} \#[y, \mathcal{S}(A, n)] = n \cdot |\mathcal{S}(A, n)|.$$

Thus, for at least one transition e_g in D, an infinite increasing sequence of nonnegative integers $\{n_k\}_{k=0}^{\infty}$ exists, such that

$$\#[e_g, \mathcal{S}(A, n_k)] \geq \frac{n_k}{|D|} \cdot |\mathcal{S}(A, n_k)|$$

for all k in $\mathbb{N}$. Thus, taking into account (2.12), (2.13), and (2.14), it can be easily seen that if the automaton A is equiloaded, then the constants $D_{J,H+1}^{(1)}$ and $D_{J,H+1}^{(2)}$ have to be both nonzero and equal. Moreover, the constants $D_{J,L}^{(1)}$ and $D_{J,L}^{(2)}$ have to be zero for $L > H + 1$. Thus, the coefficient at $(n \cdot s + a)^h \cdot \rho_J^{ns+a}$ is zero for $h > H$, in $\left|\varphi_a^{(1)}(n) - \varphi_a^{(2)}(n)\right|$. Thus, we have

$$\left|\varphi_a^{(1)}(n) - \varphi_a^{(2)}(n)\right| = O\left((n \cdot s + a)^H \cdot \rho_J^{ns+a}\right).$$

The correctness of the implication then clearly follows from (2.12).

We have proved both implications for both possible choices of $\mathcal{S}$. That is, the theorem is proved. □

It can be easily seen that the equivalence provided by Theorem 2.3.4 does not hold for $\mathcal{S} = \mathcal{C}_{\leq}$ and $\mathcal{S} = \mathcal{A}_{\leq}$ – it is a trivial task to construct a deterministic finite automaton accepting a finite language, such that it satisfies the alternative definition from Theorem 2.3.4, but at the same time, the automaton is not $\mathcal{S}$-equiloaded.

2.3.2 Computation of Basic Quantities via Convolutions

In this subsection, we shall derive an alternative method for computing the basic quantities $\#[e, \mathrm{Comp}(A, n)]$ and $\#[e, \mathrm{Acc}(A, n)]$, for a given transition e, and the quantities $\#[q, \mathrm{Comp}(A, n)]$ and $\#[q, \mathrm{Acc}(A, n)]$, for a given state q. We shall show that these quantities may be computed from certain convolutions. We shall use this alternative method of computation to prove Theorem 2.3.14 on asymptotic properties of the quantities $\#[e, \mathrm{Comp}(A, n)]$ and $\#[e, \mathrm{Acc}(A, n)]$, and, most importantly, to prove the characterization of weakly state-$\mathcal{C}_{=}$-equiloaded deterministic finite automata (Theorem 2.3.31 and preceeding lemmas).

Notation 2.3.5 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε. In addition to the notation $A_q = (K, \Sigma, \delta, q, F)$, where q is in K, introduced in Section 2.1, we shall use the notation $A^S = (K, \Sigma, \delta, q_0, S)$, where S is a subset of K. That is, by A^S we shall denote the automaton A with the set of accepting states changed to S.

Now, we shall state a lemma that provides an alternative method for computing the number of uses of a given *transition* in computation paths of a given length.

Lemma 2.3.6 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with $K = \{q_0, q_1, \ldots, q_{m-1}\}$, for some m in $\mathbb{N}$. Let $e = (q_j, c, q_k)$ in D be a transition, for some c in $\Sigma \cup \{\varepsilon\}$ and j, k in $\{0, 1, \ldots, m-1\}$. Then, for all n in $\mathbb{N}$, the identities

$$\#[e, \mathrm{Comp}(A, n)] = \sum_{i=0}^{n-1} |\mathrm{Acc}(A^{\{q_j\}}, i)| \cdot |\mathrm{Comp}(A_{q_k}, n-i-1)|,$$

and

$$\#[e, \mathrm{Acc}(A, n)] = \sum_{i=0}^{n-1} |\mathrm{Acc}(A^{\{q_j\}}, i)| \cdot |\mathrm{Acc}(A_{q_k}, n-i-1)|$$

hold.

Proof. We shall prove only the second identity, since the first identity is clearly its special case: obviously, $\mathrm{Comp}(A, n) = \mathrm{Acc}(A^K, n)$.

The first method how to prove the second identity, is by direct combinatorial insight. In fact, the number of uses of the transition e as an $(i+1)$-th step of accepting computation paths of length n, can be clearly expressed as

$$|\mathrm{Acc}(A^{\{q_j\}}, i)| \cdot |\mathrm{Acc}(A_{q_k}, n-i-1)|.$$

The overall number of uses of this transition in accepting computation paths of length n is then clearly the sum of this quantity for all possible i.

However, we shall also present the second proof that does not make use of combinatorial insight. As we have already noted in Section 2.1, the system of OΔEs for

$$t_0^e(n) = \#[e, \mathrm{Acc}(A, n)]$$

can be viewed either as a homogeneous system of $2m$ OΔEs in $2m$ unknown functions, or as a nonhomogeneous system of m OΔEs in m unknown functions. Up to now, we have always used the perspective of the homogeneous system of $2m$ OΔEs. However, if we express the quantity $t_0^e(n)$ by a nonhomogeneous system, we obtain the system

$$\mathbf{t}'_n = \Delta \cdot \mathbf{t}'_{n-1} + \mathbf{x}_{n-1},$$

where

$$\mathbf{t}'_n = \left(t_0^e(n), t_1^e(n), \ldots, t_{m-1}^e(n)\right)^T$$

and

$$\mathbf{x}_n = (\underbrace{0, \ldots, 0}_{j-1}, |\mathrm{Acc}(A_{q_k}, n)|, \underbrace{0, \ldots, 0}_{m-j})^T. \tag{2.15}$$

The initial conditions are given by

$$\mathbf{t}'_0 = (\underbrace{0, 0, \ldots, 0}_{m})^T.$$

Thus, it is clear that the column vector $\mathbf{t}'_n$ can be expressed from this nonhomogeneous system of OΔEs as

$$\mathbf{t}'_n = \mathbf{I}_m \cdot \mathbf{x}_{n-1} + \Delta \cdot \mathbf{x}_{n-2} + \ldots + \Delta^{n-2} \cdot \mathbf{x}_1 + \Delta^{n-1} \cdot \mathbf{x}_0 = \sum_{i=0}^{n-1} \Delta^i \cdot \mathbf{x}_{n-i-1}. \tag{2.16}$$

Thus, if we introduce the notation

$$\Delta^i = \begin{pmatrix} d_{0,0}^{(i)} & d_{0,1}^{(i)} & \ldots & d_{0,m-1}^{(i)} \\ d_{1,0}^{(i)} & d_{1,1}^{(i)} & \ldots & d_{1,m-1}^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m-1,0}^{(i)} & d_{m-1,1}^{(i)} & \ldots & d_{m-1,m-1}^{(i)} \end{pmatrix}$$

for i in $\mathbb{N}$, from (2.15) and (2.16) we obtain the identity

$$\#[e, \mathrm{Acc}(A, n)] = t_0^e(n) = \sum_{i=0}^{n-1} d_{0,j}^{(i)} \cdot |\mathrm{Acc}(A_{q_k}, n-i-1)|. \tag{2.17}$$

However, if we define

$$\mathbf{y} = (\underbrace{0, \ldots, 0}_{j-1}, 1, \underbrace{0, \ldots, 0}_{m-j})^T,$$

then $d_{0,j}^{(i)}$ is clearly the first entry of the column vector $\Delta^i \cdot \mathbf{y}$, i.e., by Theorem 2.1.4,

$$d_{0,j}^{(i)} = |\mathrm{Acc}(A^{\{q_j\}}, i)|.$$

Thus, (2.17) can be rewritten as

$$\#[e, \mathrm{Acc}(A, n)] = t_0^e(n) = \sum_{i=0}^{n-1} |\mathrm{Acc}(A^{\{q_j\}}, i)| \cdot |\mathrm{Acc}(A_{q_k}, n-i-1)|.$$

That is, the lemma is proved. □

The following lemma provides us with an alternative method for computing the quantities $\#[q, \mathrm{Comp}(A, n)]$ and $\#[q, \mathrm{Acc}(A, n)]$, counting the number of uses of a specified *state*. The lemma may be proved analogously as Lemma 2.3.6, or can be proved as a corollary of this lemma. We shall present the proof using the second approach.

Lemma 2.3.7 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with $K = \{q_0, q_1, \ldots, q_{m-1}\}$, for some m in $\mathbb{N}$. Let q_j in K be a state, for some j in $\{0, 1, \ldots, m-1\}$. Then, for all n in $\mathbb{N}$, the identities

$$\#[q_j, \mathrm{Comp}(A, n)] = \sum_{i=0}^{n} |\mathrm{Acc}(A^{\{q_j\}}, i)| \cdot |\mathrm{Comp}(A_{q_j}, n-i)|,$$

and

$$\#[q_j, \mathrm{Acc}(A, n)] = \sum_{i=0}^{n} |\mathrm{Acc}(A^{\{q_j\}}, i)| \cdot |\mathrm{Acc}(A_{q_j}, n-i)|$$

hold.

Proof. Once again, the lemma can be proved by a direct combinatorial insight. However, we shall present also a proof based on the use of Lemma 2.3.6. We shall prove only the second identity, since the first one is clearly its special case.

Clearly, the equation

$$\#[q_j, \mathrm{Acc}(A, n)] = |\mathrm{Acc}(A^{\{q_j\}\cap F}, n)| + \sum_{(q_j, c, q_k) \in D} \#[(q_j, c, q_k), \mathrm{Acc}(A, n)] \tag{2.18}$$

holds, where the sum goes through all c in $\Sigma \cup \{\varepsilon\}$ and all q_k in K. Then, by Lemma 2.3.6, the equation (2.18) becomes

$$\begin{aligned}
\#[q_j, \mathrm{Acc}(A, n)] &= |\mathrm{Acc}(A^{\{q_j\}\cap F}, n)| + \sum_{(q_j, c, q_k) \in D} \sum_{i=0}^{n-1} |\mathrm{Acc}(A^{\{q_j\}}, i)| \cdot |\mathrm{Acc}(A_{q_k}, n-i-1)| = \\
&= |\mathrm{Acc}(A^{\{q_j\}\cap F}, n)| + \sum_{i=0}^{n-1} \left(|\mathrm{Acc}(A^{\{q_j\}}, i)| \cdot \sum_{(q_j, c, q_k) \in D} |\mathrm{Acc}(A_{q_k}, n-i-1)| \right) = \\
&= |\mathrm{Acc}(A^{\{q_j\}\cap F}, n)| + \sum_{i=0}^{n-1} |\mathrm{Acc}(A^{\{q_j\}}, i)| \cdot |\mathrm{Acc}(A_{q_j}, n-i)| = \\
&= \sum_{i=0}^{n} |\mathrm{Acc}(A^{\{q_j\}}, i)| \cdot |\mathrm{Acc}(A_{q_j}, n-i)|,
\end{aligned}$$

where the next to the last step is by Theorem 2.1.4, and the last step makes use of the obvious fact that $|\mathrm{Acc}(A^{\{q_j\}\cap F}, n)| > 0$ implies $|\mathrm{Acc}(A_{q_j}, 0)| = 1$. Thus, the theorem is proved. □

2.3.3 Asymptotic Properties of Quantities for Strongly Connected Automata

In this subsection, we shall focus on one remarkable property of deterministic finite automata with strongly connected graphical representation. This property is stated in Theorem 2.3.14 and, in the case of automata with strongly connected graphical representation, significantly simplifies the asymptotic estimates for the quantities representing the number of uses of a given transition. This theorem also implies that a similar property holds also for the quantities representing the number of uses of a given state (Corollary 2.3.15).

However, before we proceed to the statement and proof of Theorem 2.3.14, let us briefly introduce the concept of the *period* of a given strongly connected automaton. The period of an automaton is essentialy the same as the period of its transition matrix (see, e.g., [31]). We shall present three alternative characterizations of this concept. We shall omit proofs of Lemma 2.3.8 and of Theorem 2.3.10. The reason for this is that the proofs are lengthy and technical, however they are essentially the same as the proofs of analogous results in the theory of nonnegative matrices (see, e.g., [31]).

Lemma 2.3.8 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with strongly connected graphical representation. Then, a positive integer P_A in $\mathbb{N}^+$ exists, such that K can be partitioned into P_A disjoint sets

$$\mathscr{P}(0), \mathscr{P}(1), \ldots, \mathscr{P}(P_A - 1),$$

such that

(i) If, for some q in K, w in Σ^* and n in $\mathbb{N}$, the property

$$(q_0, w) \vdash^n (q, \varepsilon)$$

holds, then q is in $\mathscr{P}(n \bmod P_A)$.

(ii) A nonnegative integer n_0 in $\mathbb{N}$ exists, such that for all $n \geq n_0$ and all q in $\mathscr{P}(n \bmod P_A)$, a word w in Σ^* exists, such that the property

$$(q_0, w) \vdash^n (q, \varepsilon)$$

holds.

Definition 2.3.9 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with strongly connected graphical representation. We shall call the positive integer P_A from the previous lemma the *period* of the automaton A.

Theorem 2.3.10 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with strongly connected graphical representation. Then, the period P_A of the automaton A is equal to the number of distinct eigenvalues of Δ with the absolute value equal to the spectral radius $\rho(\Delta)$ (i.e., their absolute value is equal to the Perron-Frobenius eigenvalue ρ). This is further equal to the greatest common divisor of lengths of closed directed walks in the graphical representation of the automaton A.

Now we are prepared to proceed to our study of asymptotics for the number of uses of a given transition. However, before we state Theorem 2.3.14, let us prove one lemma that we shall use in its proof.

Lemma 2.3.11 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with strongly connected graphical representation. Let $S, T \subseteq \mathscr{P}(k)$ be nonempty sets, for some fixed k in $\{0, 1, \ldots, P_A - 1\}$. Then

$$\left|\mathrm{Acc}(A^S, n)\right| = c \cdot \left|\mathrm{Acc}(A^T, n)\right| \pm o\left(\max\left\{1, \left|\mathrm{Acc}(A^S, n)\right|\right\}\right),$$

for some real constant c in $\mathbb{R}$.

Proof. If the automaton A consists of one isolated state without any transition, the statement of the lemma is trivial. Thus, let us suppose that A has at least one transition. Then, since the graphical representation of the automaton A is strongly connected, the transition matrix Δ is irreducible. Thus, by the Perron-Frobenius Theorem and by the results obtained in Section 2.1, we have

$$\left|\mathrm{Acc}(A^S, n)\right| = \left(\sum_{j=0}^{p-1} c_j \cdot e^{2\pi i n j/p}\right) \cdot \rho^n + o(\rho^n)$$

and

$$\left|\mathrm{Acc}(A^T, n)\right| = \left(\sum_{j=0}^{p-1} d_j \cdot e^{2\pi i n j/p}\right) \cdot \rho^n + o(\rho^n),$$

where ρ is the Perron-Frobenius eigenvalue of the transition matrix Δ, and where $c_0, \ldots, c_{p-1}$ and $d_0, \ldots, d_{p-1}$ are (in general complex) constants. Further, by Theorem 2.3.10, p is equal to the period P_A of the automaton A.

The functions

$$\sum_{j=0}^{p-1} c_j \cdot e^{2\pi i n j/p} \qquad \text{and} \qquad \sum_{j=0}^{p-1} d_j \cdot e^{2\pi i n j/p}$$

are clearly periodic with period $p = P_A$. Moreover, in the proof of Lemma 2.3.1, we have observed that these functions are real.

Now, since both S and T are subsets of $\mathscr{P}(k)$, it follows that both $|\mathrm{Acc}(A^S, n)|$ and $|\mathrm{Acc}(A^T, n)|$ are zero for n such that $n \bmod p \neq k$. Moreover, since both S and T are nonempty, it follows from Lemma 2.3.2 that both

$$\sum_{j=0}^{p-1} c_j \cdot e^{2\pi i k j/p} \qquad \text{and} \qquad \sum_{j=0}^{p-1} d_j \cdot e^{2\pi i k j/p}$$

are nonzero. Thus, clearly, the statement of the lemma holds for the real constant c defined by

$$c = \frac{\sum_{j=0}^{p-1} c_j \cdot e^{2\pi ikj/p}}{\sum_{j=0}^{p-1} d_j \cdot e^{2\pi ikj/p}}.$$

Thus, the lemma is proved. □

Before we use this lemma to prove the main result of this subsection, Theorem 2.3.14, let us prove one auxiliary lemma and its corollary that we shall use in the proof of Theorem 2.3.14.

Lemma 2.3.12 Let $\{a_n\}_{n=0}^{\infty}$, $\{b_n\}_{n=0}^{\infty}$ be sequences of real numbers, such that $a_n \geq 0, b_n > 0$ for all $n \geq n_0$, where n_0 is in $\mathbb{N}$, $b_n \to \infty$ for $n \to \infty$, and

$$\lim_{n\to\infty} \frac{a_n}{b_n} = L,$$

where L is in $\mathbb{R}$.[10] Then also

$$\lim_{n\to\infty} \frac{\sum_{k=0}^{n} a_k}{\sum_{k=0}^{n} b_k} = L.$$

Proof. By the definition of limit, it follows that for every $\varepsilon > 0$, a nonnegative integer N_ε in $\mathbb{N}$ exists (without loss of generality, let $N_\varepsilon \geq n_0$ for all $\varepsilon > 0$), such that for all $n \geq N_\varepsilon$,

$$L - \varepsilon < \frac{a_n}{b_n} < L + \varepsilon,$$

i.e.,

$$Lb_n - \varepsilon b_n < a_n < Lb_n + \varepsilon b_n.$$

Thus, for given $\varepsilon > 0$ and for $n \geq N_\varepsilon$, we have

$$L \cdot \left(\sum_{k=N_\varepsilon}^{n} b_k\right) - \varepsilon \cdot \left(\sum_{k=N_\varepsilon}^{n} b_k\right) < \sum_{k=N_\varepsilon}^{n} a_k < L \cdot \left(\sum_{k=N_\varepsilon}^{n} b_k\right) + \varepsilon \cdot \left(\sum_{k=N_\varepsilon}^{n} b_k\right),$$

i.e.,

$$L - \varepsilon < \frac{\sum_{k=N_\varepsilon}^{n} a_k}{\sum_{k=N_\varepsilon}^{n} b_k} < L + \varepsilon.$$

That is,

$$\lim_{n\to\infty} \frac{\sum_{k=N_\varepsilon}^{n} a_k}{\sum_{k=N_\varepsilon}^{n} b_k} = L.$$

However, since $b_n \to \infty$ for $n \to \infty$,

$$\lim_{n\to\infty} \frac{\sum_{k=0}^{n} a_k}{\sum_{k=0}^{n} b_k} = \lim_{n\to\infty} \frac{\sum_{k=0}^{N_\varepsilon-1} a_k + \sum_{k=N_\varepsilon}^{n} a_k}{\sum_{k=0}^{N_\varepsilon-1} b_k + \sum_{k=N_\varepsilon}^{n} b_k} = \lim_{n\to\infty} \frac{\frac{\sum_{k=0}^{N_\varepsilon-1} a_k}{\sum_{k=N_\varepsilon}^{n} b_k} + \frac{\sum_{k=N_\varepsilon}^{n} a_k}{\sum_{k=N_\varepsilon}^{n} b_k}}{\frac{\sum_{k=0}^{N_\varepsilon-1} b_k}{\sum_{k=N_\varepsilon}^{n} b_k} + 1} = \lim_{n\to\infty} \frac{\sum_{k=N_\varepsilon}^{n} a_k}{\sum_{k=N_\varepsilon}^{n} b_k} = L.$$

Thus, the lemma is proved. □

Corollary 2.3.13 Let $\{a_n\}_{n=0}^{\infty}$, $\{b_n\}_{n=0}^{\infty}$ be sequences of real numbers, such that $b_n \neq 0$ for all $n \geq n_0$, where n_0 is in $\mathbb{N}$, $b_n \to \infty$ for $n \to \infty$, and

$$a_n = o(b_n).$$

Then also

$$\sum_{k=0}^{n} a_k = o\left(\sum_{k=0}^{n} b_k\right).$$

[10] The statement holds also for $L = \pm\infty$, but the case where L is in $\mathbb{R}$ is sufficient for the purposes of this report.

Proof. The claim is a special case of the statement of Lemma 2.3.12, where $L = 0$. □

Now we shall proceed to the main result of this subsection. For deterministic finite automata without ε-transitions, and for the case of *accepting* computation paths, we have proved a similar result already in [25]. However, the proof presented in [25] is lengthy, technical, and complicated. The proof presented in what follows is considerably simpler. Thus, the presented theorem may be viewed as a generalization of the result presented in [25], and its proof may be viewed as a significant simplification of the proof from [25].

Theorem 2.3.14 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with strongly connected graphical representation. Then for each transition e in D, a real constant a_e in $\mathbb{R}$ exists, such that

$$\#[e, \mathrm{Comp}(A, n)] = (a_e n \pm O(1)) \cdot |\mathrm{Comp}(A, n)|,$$

and

$$\#[e, \mathrm{Acc}(A, n)] = (a_e n \pm O(1)) \cdot |\mathrm{Acc}(A, n)|.$$

Proof. To prove the theorem, it clearly suffices to show that a real constant a_e in $\mathbb{R}$ exists, such that

$$\#[e, \mathrm{Acc}(A^S, n)] = (a_e n \pm O(1)) \cdot |\mathrm{Acc}(A^S, n)| \tag{2.19}$$

for all sets of states $S \subseteq K$. If S is empty, then both $|\mathrm{Acc}(A^S, n)|$ and $\#[e, \mathrm{Acc}(A^S, n)]$ are zero and the property is trivially satisfied for all constants a_e. Thus, we shall assume that S is nonempty.

We shall suppose that A has at least one transition – otherwise, the claim is trivial. Then, since the graphical representation of the automaton A is strongly connected, the transition matrix Δ is irreducible. Thus, it follows from the results obtained in Section 2.1, from Theorem 2.3.10, and from Lemma 2.3.2, that

$$\#[e, \mathrm{Acc}(A^S, n)] = \left(\sum_{j=0}^{P_A - 1} c_j \cdot e^{2\pi i n j / P_A} \right) \cdot n \cdot \rho^n \pm O(1) \cdot |\mathrm{Acc}(A^S, n)|,$$

where ρ is the Perron-Frobenius eigenvalue of the transition matrix Δ, and where $c_0, \ldots, c_{P_A - 1}$ in $\mathbb{C}$ are constants. Moreover, as we have observed in the proof of Lemma 2.3.1, the periodic function

$$\sum_{j=0}^{P_A - 1} c_j \cdot e^{2\pi i n j / P_A}$$

is always real, and, by Lemma 2.3.2, $|\mathrm{Acc}(A^S, n_k)| = \Theta(\rho^{n_k})$, where $\{n_k\}_{k=0}^{\infty}$ is the infinite increasing sequence of all nonnegative integers n, such that $|\mathrm{Acc}(A^S, n)|$ is nonzero (such a sequence exists, since the graphical representation of A is strongly connected, at least one transition exists, and the set S is nonempty). Thus, we may restate this as follows: a sequence $\{b_n^S\}_{n=0}^{\infty}$ periodic with period P_A exists, such that

$$\#[e, \mathrm{Acc}(A^S, n)] = (b_n^S n \pm O(1)) \cdot |\mathrm{Acc}(A^S, n)|. \tag{2.20}$$

We shall first prove that a real constant a_e in $\mathbb{R}$ exists, such that (2.19) holds for all sets S, such that $S \subseteq \mathscr{P}(k)$ for some fixed k in $\{0, 1, \ldots, P_A - 1\}$. In that case, $|\mathrm{Acc}(A^S, n)|$ is zero for all n in $\mathbb{N}$, such that $n \bmod P_A \neq k$. Thus, clearly, for every such given S, the property

$$\#[e, \mathrm{Acc}(A^S, n)] = (b_k^S n \pm O(1)) \cdot |\mathrm{Acc}(A^S, n)|$$

holds. We shall prove that b_k^S is the same for all such S, and that will be our constant a_e.

First, we shall prove that $b_0^S = b_0^T$ for all nonempty $S, T \subseteq \mathscr{P}(0)$. By Lemma 2.3.11,

$$|\mathrm{Acc}(A^S, n)| = c \cdot \left| \mathrm{Acc}(A^T, n) \right| \pm o\left(\max\left\{1, \left| \mathrm{Acc}(A^S, n) \right| \right\} \right). \tag{2.21}$$

By Lemma 2.3.6, and subsequently by (2.21), Corollary 2.3.13, and once again Lemma 2.3.6,

$$\begin{aligned}
\#[e, \mathrm{Acc}(A^S, n)] &= \sum_{i=0}^{n-1} |\mathrm{Acc}(A^{\{q_j\}}, i)| \cdot |\mathrm{Acc}(A^S_{q_k}, n-i-1)| = \\
&= \sum_{i=0}^{n-1} |\mathrm{Acc}(A^{\{q_j\}}, i)| \cdot \Big(c \cdot |\mathrm{Acc}(A^T_{q_k}, n-i-1)| \pm \\
&\qquad\qquad \pm o\left(\max\left\{1, |\mathrm{Acc}(A^S_{q_k}, n-i-1)|\right\}\right)\Big) = \\
&= \left(\sum_{i=0}^{n-1} c \cdot |\mathrm{Acc}(A^{\{q_j\}}, i)| \cdot |\mathrm{Acc}(A^T_{q_k}, n-i-1)|\right) \pm O(1) \cdot |\mathrm{Acc}(A^S, n)| = \\
&= \left(c \cdot \sum_{i=0}^{n-1} |\mathrm{Acc}(A^{\{q_j\}}, i)| \cdot |\mathrm{Acc}(A^T_{q_k}, n-i-1)|\right) \pm O(1) \cdot |\mathrm{Acc}(A^S, n)| = \\
&= c \cdot \#[e, \mathrm{Acc}(A^T, n)] \pm O(1) \cdot |\mathrm{Acc}(A^S, n)|.
\end{aligned}$$

This clearly implies that $b_0^S = b_0^T =: b_0$.

Now, let $T \subseteq \mathscr{P}(k)$ for some k in $\{1, 2, \ldots, P_A - 1\}$. We shall show that $b_k^T = b_0$. By what we have proved up to now, it clearly suffices to show that $b_k^{\mathscr{P}(k)} = b_0$. However,

$$\begin{aligned}
\#[e, \mathrm{Acc}(A^{\mathscr{P}(k)}, n)] &= \sum_{q \in \mathscr{P}(0)} \Big(\#[e, \mathrm{Acc}(A^{\{q\}}, n-k)] \cdot |\mathrm{Acc}(A_q^{\mathscr{P}(k)}, k)| + \\
&\qquad + |\mathrm{Acc}(A^{\{q\}}, n-k)| \cdot \#[e, \mathrm{Acc}(A_q^{\mathscr{P}(k)}, k)]\Big) = \\
&= \sum_{q \in \mathscr{P}(0)} \Big(\#[e, \mathrm{Acc}(A^{\{q\}}, n-k)] \cdot |\mathrm{Acc}(A_q^{\mathscr{P}(k)}, k)|\Big) + \\
&\quad + \sum_{q \in \mathscr{P}(0)} \Big(|\mathrm{Acc}(A^{\{q\}}, n-k)| \cdot \#[e, \mathrm{Acc}(A_q^{\mathscr{P}(k)}, k)]\Big). \qquad (2.22)
\end{aligned}$$

Moreover, clearly,

$$|\mathrm{Acc}(A^{\mathscr{P}(k)}, n)| = \sum_{q \in \mathscr{P}(0)} |\mathrm{Acc}(A^{\{q\}}, n-k)| \cdot |\mathrm{Acc}(A_q^{\mathscr{P}(k)}, k)|.$$

Thus, by what we have proved above,

$$\begin{aligned}
&\sum_{q \in \mathscr{P}(0)} \Big(\#[e, \mathrm{Acc}(A^{\{q\}}, n-k)] \cdot |\mathrm{Acc}(A_q^{\mathscr{P}(k)}, k)|\Big) = \\
&\quad = \sum_{q \in \mathscr{P}(0)} \Big((b_0(n-k) \pm O(1)) \cdot |\mathrm{Acc}(A^{\{q\}}, n-k)| \cdot |\mathrm{Acc}(A_q^{\mathscr{P}(k)}, k)|\Big) = \\
&\quad = \sum_{q \in \mathscr{P}(0)} \Big((b_0 n \pm O(1)) \cdot |\mathrm{Acc}(A^{\{q\}}, n-k)| \cdot |\mathrm{Acc}(A_q^{\mathscr{P}(k)}, k)|\Big) = \\
&\quad = (b_0 n \pm O(1)) \cdot \sum_{q \in \mathscr{P}(0)} |\mathrm{Acc}(A^{\{q\}}, n-k)| \cdot |\mathrm{Acc}(A_q^{\mathscr{P}(k)}, k)| = \\
&\quad = (b_0 n \pm O(1)) \cdot |\mathrm{Acc}(A^{\mathscr{P}(k)}, n)|. \qquad (2.23)
\end{aligned}$$

Moreover, since $\#[e, \mathrm{Acc}(A_q^{\mathscr{P}(k)}, k)]$ is a constant that is zero whenever $|\mathrm{Acc}(A_q^{\mathscr{P}(k)}, k)|$ is zero, we have

$$\sum_{q\in\mathscr{P}(0)}\left(|\mathrm{Acc}(A^{\{q\}},n-k)|\cdot\#[e,\mathrm{Acc}(A_q^{\mathscr{P}(k)},k)]\right)=$$
$$=O\left(\sum_{q\in\mathscr{P}(0)}|\mathrm{Acc}(A^{\{q\}},n-k)|\cdot|\mathrm{Acc}(A_q^{\mathscr{P}(k)},k)|\right)=$$
$$=O(|\mathrm{Acc}(A^{\mathscr{P}(k)},n)|)=O(1)\cdot|\mathrm{Acc}(A^{\mathscr{P}(k)},n)|. \tag{2.24}$$

Now, by plugging (2.23) and (2.24) into (2.22), we obtain

$$\#[e,\mathrm{Acc}(A^{\mathscr{P}(k)},n)]=(b_0n\pm O(1))\cdot|\mathrm{Acc}(A^{\mathscr{P}(k)},n)|.$$

Thus, we have proved that for all sets of states T, such that $T\subseteq\mathscr{P}(k)$ for some k in the set $\{0,1,\ldots,P_A-1\}$, the property

$$\#[e,\mathrm{Acc}(A^T,n)]=(b_0n\pm O(1))\cdot|\mathrm{Acc}(A^T,n)|.$$

holds. It remains to prove that the property holds also for all other sets of states S. However, this is clear, since for n such that $n \bmod P_A=k$,

$$\#[e,\mathrm{Acc}(A^S,n)]=\#[e,\mathrm{Acc}(A^{S\cap\mathscr{P}(k)},n)]=(b_0n\pm O(1))\cdot|\mathrm{Acc}(A^{S\cap\mathscr{P}(k)},n)|=$$
$$=(b_0n\pm O(1))\cdot|\mathrm{Acc}(A^S,n)|.$$

Since this holds for all k in $\{0,1,\ldots,P_A-1\}$, the theorem is proved. □

Theorem 2.3.14 directly implies the following corollary for the quantities representing the numbers of uses of states.

Corollary 2.3.15 Let $A=(K,\Sigma,\delta,q_0,F)$ be a DFAε with strongly connected graphical representation. Then for each state q in K, a real constant a_q in $\mathbb{R}$ exists, such that

$$\#[q,\mathrm{Comp}(A,n)]=(a_qn\pm O(1))\cdot|\mathrm{Comp}(A,n)|,$$

and

$$\#[q,\mathrm{Acc}(A,n)]=(a_qn\pm O(1))\cdot|\mathrm{Acc}(A,n)|.$$

Proof. In Theorem 2.1.11, we have observed that the identities

$$\#[q,\mathrm{Comp}(A,n)]=\sum_{(p,c,q)\in D}\#[(p,c,q),\mathrm{Comp}(A,n)],$$
$$\#[q,\mathrm{Acc}(A,n)]=\sum_{(p,c,q)\in D}\#[(p,c,q),\mathrm{Acc}(A,n)]$$

hold for all states $q\neq q_0$, and that for the state q_0, the identities

$$\#[q_0,\mathrm{Comp}(A,n)]=|\mathrm{Comp}(A,n)|+\sum_{(p,c,q_0)\in D}\#[(p,c,q_0),\mathrm{Comp}(A,n)],$$
$$\#[q_0,\mathrm{Acc}(A,n)]=|\mathrm{Acc}(A,n)|+\sum_{(p,c,q_0)\in D}\#[(p,c,q_0),\mathrm{Acc}(A,n)]$$

hold (the sums go through all p in K and c in $\Sigma\cup\{\varepsilon\}$). Thus, as a direct consequence, the claim holds for

$$a_q=\sum_{(p,c,q)\in D}a_{(p,c,q)}.$$

That is, the corollary is proved. □

2.3.4 Almost-Equivalence of Transition-Equiloadedness $\mathcal{S}$-Measures

In this subsection, we shall use Theorem 2.3.14 to prove Theorem 2.3.16 that will be of key importance for the theory of transition-equiloaded DFA and DFAε. We shall prove that the transition-equiloadedness $\mathcal{S}$-measures are *almost* equivalent for $\mathcal{S}$ in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$. These measures are not equivalent only for DFAε that do not contain any reachable directed cycle from which at least one accepting state is reachable, i.e., for a subset of DFAε accepting finite languages.

This fact can be viewed as a justification of the definition of transition-equiloadedness, since it shows a certain robustness of this definition – no matter which of the standard parameters $\mathcal{S}$ is chosen, there are only minor differences in the resulting family of automata and languages.

However, for the case of state-equiloadedness, this property does not hold. We shall prove that the values of state-equiloadedness $\mathcal{S}$-measures may significantly vary for different $\mathcal{S}$ also for automata, for which the measures of transition-equiloadedness have to be the same.

Theorem 2.3.16 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε, such that in the graphical representation of A, there is at least one reachable directed cycle from which an accepting state is reachable. Then,

$$B_A(\mathcal{C}_=) = B_A(\mathcal{A}_=) = B_A(\mathcal{C}_\leq) = B_A(\mathcal{A}_\leq).$$

Proof. First, let us suppose that the graphical representation of the automaton A is *not* strongly connected. Then, there is at least one transition e in D, such that

$$\#[e, \gamma] \leq 1$$

for all computation paths γ. Thus,

$$\#[e, \mathrm{Comp}(A, n)] = O(1) \cdot |\mathrm{Comp}(A, n)| \tag{2.25}$$

and

$$\#[e, \mathrm{Acc}(A, n)] = O(1) \cdot |\mathrm{Acc}(A, n)|. \tag{2.26}$$

On the other hand, clearly,

$$\sum_{y \in D} \#[y, \mathrm{Comp}(A, n)] = n \cdot |\mathrm{Comp}(A, n)|$$

and

$$\sum_{y \in D} \#[y, \mathrm{Acc}(A, n)] = n \cdot |\mathrm{Acc}(A, n)|.$$

Thus, transitions f_1, f_2 in D and infinite increasing sequences $\{n_k\}_{k=0}^{\infty}$, $\{m_k\}_{k=0}^{\infty}$ of nonnegative integers have to exist (since there is a reachable directed cycle in the graphical representation of A, from which an accepting state is reachable), such that

$$\begin{aligned} |\mathrm{Comp}(A, n_k)| &> 0, \\ |\mathrm{Acc}(A, m_k)| &> 0, \\ \#[f_1, \mathrm{Comp}(A, n_k)] &\geq \frac{n_k}{|D|} \cdot |\mathrm{Comp}(A, n_k)|, \end{aligned}$$

and

$$\#[f_2, \mathrm{Acc}(A, m_k)] \geq \frac{m_k}{|D|} \cdot |\mathrm{Acc}(A, m_k)|$$

for all k in $\mathbb{N}$. Thus, by Lemma 1.5.3,

$$B_A(\mathcal{C}_=) \leq \liminf_{n \to \infty} \frac{\#[e, \mathrm{Comp}(A, n)] + 1}{\#[f_1, \mathrm{Comp}(A, n)] + 1} \leq \liminf_{k \to \infty} \frac{\#[e, \mathrm{Comp}(A, n_k)] + 1}{\#[f_1, \mathrm{Comp}(A, n_k)] + 1} = 0,$$

and

$$B_A(\mathcal{A}_=) \leq \liminf_{n\to\infty} \frac{\#[e, \mathrm{Acc}(A,n)] + 1}{\#[f_2, \mathrm{Acc}(A,n)] + 1} \leq \liminf_{k\to\infty} \frac{\#[e, \mathrm{Acc}(A,m_k)] + 1}{\#[f_2, \mathrm{Acc}(A,m_k)] + 1} = 0.$$

Thus, $B_A(\mathcal{C}_=) = B_A(\mathcal{A}_=) = 0$. Moreover, clearly,

$$\#[e, \mathrm{Comp}(A,n_k)] = o\left(\#[f_1, \mathrm{Comp}(A,n_k)]\right),$$

and

$$\#[e, \mathrm{Acc}(A,m_k)] = o\left(\#[f_2, \mathrm{Acc}(A,m_k)]\right).$$

Thus, by Corollary 2.3.13,

$$\sum_{j=0}^{k} \#[e, \mathrm{Comp}(A,n_j)] = o\left(\sum_{j=0}^{k} \#[f_1, \mathrm{Comp}(A,n_j)]\right), \tag{2.27}$$

and

$$\sum_{j=0}^{k} \#[e, \mathrm{Acc}(A,m_j)] = o\left(\sum_{j=0}^{k} \#[f_2, \mathrm{Acc}(A,m_j)]\right). \tag{2.28}$$

Now, let us assume that the sequences $\{n_k\}_{k=0}^{\infty}$ and $\{m_k\}_{k=0}^{\infty}$ contain *all* nonnegative integers, such that the conditions imposed on them are satisfied. Then, by what we have observed in the proof of Theorem 2.3.4, for some k_0 in $\mathbb{N}$, the sequences $\{n_k\}_{k=k_0}^{\infty}$ and $\{m_k\}_{k=k_0}^{\infty}$ are periodic with period s in $\mathbb{N}$. Thus, adding terms $\#[e, \mathrm{Comp}(A,l)]$ resp. $\#[e, \mathrm{Acc}(A,l)]$ for l not in $\{n_k\}_{k=0}^{\infty}$ resp. $\{m_k\}_{k=0}^{\infty}$ to the sum on the left side of (2.27) resp. (2.28) will preserve the asymptotic relation. Thus, we obtain

$$\#[e, \mathrm{Comp}(A, \leq n_k)] = o\left(\sum_{j=0}^{k} \#[f_1, \mathrm{Comp}(A,n_j)]\right),$$

and

$$\#[e, \mathrm{Acc}(A, \leq m_k)] = o\left(\sum_{j=0}^{k} \#[f_2, \mathrm{Acc}(A,m_j)]\right),$$

and, as a direct consequence,

$$\#[e, \mathrm{Comp}(A, \leq n)] + 1 = o\left(\#[f_1, \mathrm{Comp}(A, \leq n)] + 1\right),$$

and

$$\#[e, \mathrm{Acc}(A, \leq n)] + 1 = o\left(\#[f_2, \mathrm{Acc}(A, \leq n)] + 1\right).$$

But this is equivalent to that $B_A(\mathcal{C}_\leq) = B_A(\mathcal{A}_\leq) = 0$. Thus, we have proved that if the graphical representation of the automaton A is not strongly connected, then the property

$$B_A(\mathcal{C}_=) = B_A(\mathcal{A}_=) = B_A(\mathcal{C}_\leq) = B_A(\mathcal{A}_\leq) = 0$$

holds.

Now, let us suppose that the graphical representation of the automaton A *is* strongly connected. Then, by Theorem 2.3.14, for each transition e in D a constant a_e in $\mathbb{R}$ exists, such that

$$\#[e, \mathrm{Comp}(A,n)] = (a_e n \pm O(1)) \cdot |\mathrm{Comp}(A,n)|,$$

and

$$\#[e, \mathrm{Acc}(A,n)] = (a_e n \pm O(1)) \cdot |\mathrm{Acc}(A,n)|.$$

Let us denote

$$B = \min_{(e,f)\in D^2} \frac{a_e}{a_f}.$$

By Lemma 1.5.3,

$$B_A(\mathcal{C}_=) = \min_{(e,f)\in D^2} \liminf_{n\to\infty} \frac{\#[e,\mathrm{Comp}(A,n)]+1}{\#[f,\mathrm{Comp}(A,n)]+1} =$$
$$= \min_{(e,f)\in D^2} \liminf_{n\to\infty} \frac{(a_e n \pm O(1))|\mathrm{Comp}(A,n)|+1}{(a_f n \pm O(1))|\mathrm{Comp}(A,n)|+1} = \min_{(e,f)\in D^2} \frac{a_e}{a_f} = B,$$

and

$$B_A(\mathcal{A}_=) = \min_{(e,f)\in D^2} \liminf_{n\to\infty} \frac{\#[e,\mathrm{Acc}(A,n)]+1}{\#[f,\mathrm{Acc}(A,n)]+1} =$$
$$= \min_{(e,f)\in D^2} \liminf_{n\to\infty} \frac{(a_e n \pm O(1))|\mathrm{Acc}(A,n)|+1}{(a_f n \pm O(1))|\mathrm{Acc}(A,n)|+1} = \min_{(e,f)\in D^2} \frac{a_e}{a_f} = B.$$

Moreover, by Lemma 1.5.3,

$$B_A(\mathcal{C}_\le) = \min_{(e,f)\in D^2} \liminf_{n\to\infty} \frac{\#[e,\mathrm{Comp}(A,\le n)]+1}{\#[f,\mathrm{Comp}(A,\le n)]+1} = \min_{(e,f)\in D^2} \liminf_{n\to\infty} \frac{\sum_{k=0}^n \#[e,\mathrm{Comp}(A,k)]+1}{\sum_{k=0}^n \#[f,\mathrm{Comp}(A,k)]+1} =$$
$$= \min_{(e,f)\in D^2} \liminf_{n\to\infty} \frac{\sum_{k=0}^n ((a_e k \pm O(1))\cdot|\mathrm{Comp}(A,k)|)+1}{\sum_{k=0}^n ((a_f k \pm O(1))\cdot|\mathrm{Comp}(A,k)|)+1} =$$
$$= \min_{(e,f)\in D^2} \liminf_{n\to\infty} \frac{a_e\cdot\sum_{k=0}^n ((k \pm O(1))\cdot|\mathrm{Comp}(A,k)|)+1}{a_f\cdot\sum_{k=0}^n ((k \pm O(1))\cdot|\mathrm{Comp}(A,k)|)+1} = \min_{(e,f)\in D^2} \frac{a_e}{a_f} = B.$$

Similarly,

$$B_A(\mathcal{A}_\le) = \min_{(e,f)\in D^2} \liminf_{n\to\infty} \frac{\#[e,\mathrm{Acc}(A,\le n)]+1}{\#[f,\mathrm{Acc}(A,\le n)]+1} = \min_{(e,f)\in D^2} \liminf_{n\to\infty} \frac{\sum_{k=0}^n \#[e,\mathrm{Acc}(A,k)]+1}{\sum_{k=0}^n \#[f,\mathrm{Acc}(A,k)]+1} =$$
$$= \min_{(e,f)\in D^2} \liminf_{n\to\infty} \frac{\sum_{k=0}^n ((a_e k \pm O(1))\cdot|\mathrm{Acc}(A,k)|)+1}{\sum_{k=0}^n ((a_f k \pm O(1))\cdot|\mathrm{Acc}(A,k)|)+1} =$$
$$= \min_{(e,f)\in D^2} \liminf_{n\to\infty} \frac{a_e\cdot\sum_{k=0}^n ((k \pm O(1))\cdot|\mathrm{Acc}(A,k)|)+1}{a_f\cdot\sum_{k=0}^n ((k \pm O(1))\cdot|\mathrm{Acc}(A,k)|)+1} = \min_{(e,f)\in D^2} \frac{a_e}{a_f} = B.$$

Thus, we have proved that if the graphical representation of the automaton A is strongly connected, then

$$B_A(\mathcal{C}_=) = B_A(\mathcal{A}_=) = B_A(\mathcal{C}_\le) = B_A(\mathcal{A}_\le) = B.$$

That is, the theorem is proved. □

As we have already anticipated, the situation is far more complicated for the case of state-$\mathcal{S}$-equiloadedness. In the following example, we shall show an example of a deterministic finite automaton, for which the properties proved for transition-equiloadedness measures are violated for state-equiloadedness measures. To be more specific, we shall construct a deterministic finite automaton A, such that state-equiloadedness measures $\beta_A(\mathcal{C}_=), \beta_A(\mathcal{A}_=), \beta_A(\mathcal{C}_\le)$, and $\beta_A(\mathcal{A}_\le)$ attain pairwise distinct values.

Example 2.3.17 Let us consider a deterministic finite automaton $A = (K, \Sigma, \delta, q_0, F)$ defined as follows: $K = \{q_0, q_1, q_2, q_3, q_4, q_5\}$, $\Sigma = \{a, b, c, d\}$, $F = \{q_2, q_3, q_5\}$, and

$$\begin{array}{lllll}
\delta(q_0,a) = q_1, & \delta(q_0,b) = q_1, & \delta(q_0,c) = q_2, & \delta(q_0,d) = q_4, & \delta(q_1,a) = q_0, \\
\delta(q_1,b) = q_0, & \delta(q_2,a) = q_3, & \delta(q_2,b) = q_3, & \delta(q_3,a) = q_2, & \delta(q_3,b) = q_2, \\
\delta(q_4,a) = q_5, & \delta(q_5,a) = q_4, & \delta(q_5,b) = q_4, & \delta(q_5,c) = q_4, & \delta(q_5,d) = q_4.
\end{array}$$

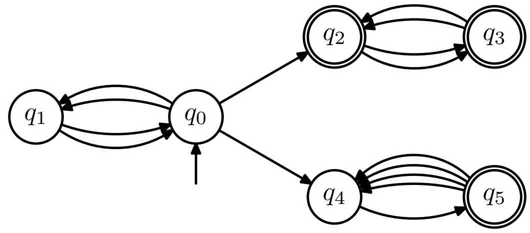


Figure 2.3: The automaton A. Since the number of transitions is relatively high, and since the characters are not important in this example, character labels are omitted in the diagram.

The graphical representation of the automaton A is depicted in Figure 2.3 (without labels at transitions specifying characters – this example works no matter what characters are used in the transitions, and there is not enough space in the figure to include these labels).

By implementing the method for computing the closed form of the basic quantities, presented in Section 2.1, on a computer, one may easily obtain the following results:

$$\begin{aligned}
|\mathrm{Comp}(A,n)| &= \frac{7}{16}\cdot n\cdot 2^n + \frac{5}{4}\cdot 2^n - \frac{1}{16}\cdot n\cdot(-2)^n - \frac{1}{4}\cdot(-2)^n,\\
|\mathrm{Acc}(A,n)| &= \frac{5}{16}\cdot n\cdot 2^n + \frac{1}{8}\cdot 2^n + \frac{1}{16}\cdot n\cdot(-2)^n - \frac{1}{8}\cdot(-2)^n,\\
|\mathrm{Comp}(A,\le n)| &= \frac{7}{8}\cdot n\cdot 2^n + \frac{13}{8}\cdot 2^n - \frac{1}{24}\cdot n\cdot(-2)^n - \frac{13}{72}\cdot(-2)^n - \frac{4}{9},\\
|\mathrm{Acc}(A,\le n)| &= \frac{5}{8}\cdot n\cdot 2^n - \frac{3}{8}\cdot 2^n + \frac{1}{24}\cdot n\cdot(-2)^n - \frac{5}{72}\cdot(-2)^n + \frac{4}{9},
\end{aligned}$$

$$\begin{aligned}
\#[q_0,\mathrm{Comp}(A,n)] &= \frac{7}{64}\cdot n^2\cdot 2^n + \frac{27}{32}\cdot n\cdot 2^n + \frac{15}{16}\cdot 2^n - \frac{1}{64}\cdot n^2\cdot(-2)^n - \frac{5}{32}\cdot n\cdot(-2)^n +\\
&\quad + \frac{1}{16}\cdot(-2)^n,\\
\#[q_1,\mathrm{Comp}(A,n)] &= \frac{7}{64}\cdot n^2\cdot 2^n + \frac{13}{32}\cdot n\cdot 2^n + \frac{3}{16}\cdot 2^n - \frac{1}{64}\cdot n^2\cdot(-2)^n - \frac{3}{32}\cdot n\cdot(-2)^n -\\
&\quad - \frac{3}{16}\cdot(-2)^n,\\
\#[q_2,\mathrm{Comp}(A,n)] &= \frac{1}{16}\cdot n^2\cdot 2^n + \frac{3}{16}\cdot n\cdot 2^n + \frac{3}{32}\cdot 2^n - \frac{1}{16}\cdot n\cdot(-2)^n - \frac{3}{32}\cdot(-2)^n,\\
\#[q_3,\mathrm{Comp}(A,n)] &= \frac{1}{16}\cdot n^2\cdot 2^n + \frac{1}{16}\cdot n\cdot 2^n - \frac{1}{32}\cdot 2^n + \frac{1}{16}\cdot n\cdot(-2)^n + \frac{1}{32}\cdot(-2)^n,\\
\#[q_4,\mathrm{Comp}(A,n)] &= \frac{3}{64}\cdot n^2\cdot 2^n + \frac{5}{32}\cdot n\cdot 2^n + \frac{3}{32}\cdot 2^n - \frac{1}{64}\cdot n^2\cdot(-2)^n - \frac{3}{32}\cdot n\cdot(-2)^n -\\
&\quad - \frac{3}{32}\cdot(-2)^n,\\
\#[q_5,\mathrm{Comp}(A,n)] &= \frac{3}{64}\cdot n^2\cdot 2^n + \frac{1}{32}\cdot n\cdot 2^n - \frac{1}{32}\cdot 2^n - \frac{1}{64}\cdot n^2\cdot(-2)^n + \frac{1}{32}\cdot n\cdot(-2)^n +\\
&\quad + \frac{1}{32}\cdot(-2)^n,
\end{aligned}$$

$$\begin{aligned}
\#[q_0,\mathrm{Acc}(A,n)] &= \frac{5}{64}\cdot n^2\cdot 2^n + \frac{7}{32}\cdot n\cdot 2^n + \frac{3}{32}\cdot 2^n + \frac{1}{64}\cdot n^2\cdot(-2)^n - \frac{1}{32}\cdot n\cdot(-2)^n -\\
&\quad - \frac{3}{32}\cdot(-2)^n,
\end{aligned}$$

$$
\begin{aligned}
\#[q_1, \mathrm{Acc}(A,n)] &= \frac{5}{64}\cdot n^2\cdot 2^n - \frac{3}{32}\cdot n\cdot 2^n - \frac{1}{32}\cdot 2^n + \frac{1}{64}\cdot n^2\cdot(-2)^n - \frac{3}{32}\cdot n\cdot(-2)^n + \\
&\quad + \frac{1}{32}\cdot(-2)^n,\\
\#[q_2, \mathrm{Acc}(A,n)] &= \frac{1}{16}\cdot n^2\cdot 2^n + \frac{3}{16}\cdot n\cdot 2^n + \frac{3}{32}\cdot 2^n - \frac{1}{16}\cdot n\cdot(-2)^n - \frac{3}{32}\cdot(-2)^n,\\
\#[q_3, \mathrm{Acc}(A,n)] &= \frac{1}{16}\cdot n^2\cdot 2^n + \frac{1}{16}\cdot n\cdot 2^n - \frac{1}{32}\cdot 2^n + \frac{1}{16}\cdot n\cdot(-2)^n + \frac{1}{32}\cdot(-2)^n,\\
\#[q_4, \mathrm{Acc}(A,n)] &= \frac{1}{64}\cdot n^2\cdot 2^n + \frac{1}{32}\cdot n\cdot 2^n + \frac{1}{64}\cdot n^2\cdot(-2)^n + \frac{1}{32}\cdot n\cdot(-2)^n,\\
\#[q_5, \mathrm{Acc}(A,n)] &= \frac{1}{64}\cdot n^2\cdot 2^n + \frac{1}{32}\cdot n\cdot 2^n + \frac{1}{64}\cdot n^2\cdot(-2)^n + \frac{1}{32}\cdot n\cdot(-2)^n,
\end{aligned}
$$

$$
\begin{aligned}
\#[q_0, \mathrm{Comp}(A,\le n)] &= \frac{7}{32}\cdot n^2\cdot 2^n + \frac{5}{4}\cdot n\cdot 2^n + \frac{27}{32}\cdot 2^n - \frac{1}{96}\cdot n^2\cdot(-2)^n - \frac{1}{9}\cdot n\cdot(-2)^n + \\
&\quad + \frac{7}{864}\cdot(-2)^n + \frac{4}{27},\\
\#[q_1, \mathrm{Comp}(A,\le n)] &= \frac{7}{32}\cdot n^2\cdot 2^n + \frac{3}{8}\cdot n\cdot 2^n + \frac{7}{32}\cdot 2^n - \frac{1}{96}\cdot n^2\cdot(-2)^n - \frac{5}{72}\cdot n\cdot(-2)^n - \\
&\quad - \frac{125}{864}\cdot(-2)^n - \frac{2}{27},\\
\#[q_2, \mathrm{Comp}(A,\le n)] &= \frac{1}{8}\cdot n^2\cdot 2^n + \frac{1}{8}\cdot n\cdot 2^n + \frac{3}{16}\cdot 2^n - \frac{1}{24}\cdot n\cdot(-2)^n - \frac{11}{144}\cdot(-2)^n - \frac{1}{9},\\
\#[q_3, \mathrm{Comp}(A,\le n)] &= \frac{1}{8}\cdot n^2\cdot 2^n - \frac{1}{8}\cdot n\cdot 2^n + \frac{3}{16}\cdot 2^n + \frac{1}{24}\cdot n\cdot(-2)^n + \frac{5}{144}\cdot(-2)^n - \frac{2}{9},\\
\#[q_4, \mathrm{Comp}(A,\le n)] &= \frac{3}{32}\cdot n^2\cdot 2^n + \frac{1}{8}\cdot n\cdot 2^n + \frac{5}{32}\cdot 2^n - \frac{1}{96}\cdot n^2\cdot(-2)^n - \frac{5}{72}\cdot n\cdot(-2)^n - \\
&\quad - \frac{71}{864}\cdot(-2)^n - \frac{2}{27},\\
\#[q_5, \mathrm{Comp}(A,\le n)] &= \frac{3}{32}\cdot n^2\cdot 2^n - \frac{1}{8}\cdot n\cdot 2^n + \frac{5}{32}\cdot 2^n - \frac{1}{96}\cdot n^2\cdot(-2)^n + \frac{1}{72}\cdot n\cdot(-2)^n + \\
&\quad + \frac{25}{864}\cdot(-2)^n - \frac{5}{27},
\end{aligned}
$$

$$
\begin{aligned}
\#[q_0, \mathrm{Acc}(A,\le n)] &= \frac{5}{32}\cdot n^2\cdot 2^n + \frac{1}{8}\cdot n\cdot 2^n + \frac{7}{32}\cdot 2^n + \frac{1}{96}\cdot n^2\cdot(-2)^n - \frac{1}{72}\cdot n\cdot(-2)^n - \\
&\quad - \frac{61}{864}\cdot(-2)^n - \frac{4}{27},\\
\#[q_1, \mathrm{Acc}(A,\le n)] &= \frac{5}{32}\cdot n^2\cdot 2^n - \frac{1}{2}\cdot n\cdot 2^n + \frac{19}{32}\cdot 2^n + \frac{1}{96}\cdot n^2\cdot(-2)^n - \frac{1}{18}\cdot n\cdot(-2)^n - \\
&\quad - \frac{1}{864}\cdot(-2)^n - \frac{16}{27},\\
\#[q_2, \mathrm{Acc}(A,\le n)] &= \frac{1}{8}\cdot n^2\cdot 2^n + \frac{1}{8}\cdot n\cdot 2^n + \frac{3}{16}\cdot 2^n - \frac{1}{24}\cdot n\cdot(-2)^n - \frac{11}{144}\cdot(-2)^n - \frac{1}{9},\\
\#[q_3, \mathrm{Acc}(A,\le n)] &= \frac{1}{8}\cdot n^2\cdot 2^n - \frac{1}{8}\cdot n\cdot 2^n + \frac{3}{16}\cdot 2^n + \frac{1}{24}\cdot n\cdot(-2)^n + \frac{5}{144}\cdot(-2)^n - \frac{2}{9},\\
\#[q_4, \mathrm{Acc}(A,\le n)] &= \frac{1}{32}\cdot n^2\cdot 2^n + \frac{1}{32}\cdot 2^n + \frac{1}{96}\cdot n^2\cdot(-2)^n + \frac{1}{36}\cdot n\cdot(-2)^n + \frac{5}{864}\cdot(-2)^n - \\
&\quad - \frac{1}{27},
\end{aligned}
$$

$$\#[q_5, \mathrm{Acc}(A, \leq n)] = \frac{1}{32} \cdot n^2 \cdot 2^n + \frac{1}{32} \cdot 2^n + \frac{1}{96} \cdot n^2 \cdot (-2)^n + \frac{1}{36} \cdot n \cdot (-2)^n + \frac{5}{864} \cdot (-2)^n - \\ - \frac{1}{27}.$$

Thus, it is possible to observe that, by Lemma 1.5.3,

$$\beta_A(\mathcal{C}_=) = \min_{(p,q) \in K^2} \liminf_{n \to \infty} \frac{\#[p, \mathrm{Comp}(A, n)] + 1}{\#[q, \mathrm{Comp}(A, n)] + 1} = \liminf_{n \to \infty} \frac{\#[q_4, \mathrm{Comp}(A, n)] + 1}{\#[q_0, \mathrm{Comp}(A, n)] + 1} = \frac{1}{3},$$

$$\beta_A(\mathcal{A}_=) = \min_{(p,q) \in K^2} \liminf_{n \to \infty} \frac{\#[p, \mathrm{Acc}(A, n)] + 1}{\#[q, \mathrm{Acc}(A, n)] + 1} = \liminf_{n \to \infty} \frac{\#[q_4, \mathrm{Acc}(A, n)] + 1}{\#[q_0, \mathrm{Acc}(A, n)] + 1} = 0,$$

$$\beta_A(\mathcal{C}_\leq) = \min_{(p,q) \in K^2} \liminf_{n \to \infty} \frac{\#[p, \mathrm{Comp}(A, \leq n)] + 1}{\#[q, \mathrm{Comp}(A, \leq n)] + 1} = \liminf_{n \to \infty} \frac{\#[q_4, \mathrm{Comp}(A, \leq n)] + 1}{\#[q_0, \mathrm{Comp}(A, \leq n)] + 1} = \frac{2}{5},$$

$$\beta_A(\mathcal{A}_\leq) = \min_{(p,q) \in K^2} \liminf_{n \to \infty} \frac{\#[p, \mathrm{Acc}(A, \leq n)] + 1}{\#[q, \mathrm{Acc}(A, \leq n)] + 1} = \liminf_{n \to \infty} \frac{\#[q_4, \mathrm{Acc}(A, \leq n)] + 1}{\#[q_0, \mathrm{Acc}(A, \leq n)] + 1} = \frac{1}{7}.$$

That is, these four state-equiloadedness $\mathcal{S}$-measures are pairwise different.

2.3.5 Characterizations of Weak $\mathcal{S}$-Equiloadedness for several $\mathcal{S}$

In this subsection, we shall prove characterizations of weakly transition-$\mathcal{S}$-equiloaded and weakly state-$\mathcal{S}$-equiloaded DFA and DFAε, for several possible choices of $\mathcal{S}$.

We shall start with the characterization of weak transition-equiloadedness for $\mathcal{S}$ in $\{\mathcal{C}_=, \mathcal{A}_=\}$. In [25], we have presented the characterization of weak transition-$\mathcal{A}_=$-equiloadedness for DFA (without ε-transitions). The proof presented in this report is based on the proof from [25], however is slightly different. The remaining characterizations presented in this subsection are entirely new.

Theorem 2.3.18 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with connected graphical representation.

a) A is weakly transition-$\mathcal{C}_=$-equiloaded, if and only if its graphical representation either does not contain any reachable directed cycle, or it is strongly connected.

b) A is weakly transition-$\mathcal{A}_=$-equiloaded, if and only if its graphical representation either does not contain any reachable directed cycle from which some accepting state is reachable, or it is strongly connected.

Proof. First, we shall prove the easier left-to-right implications (i.e., the *only if* part). Let the automaton A be weakly transition-$\mathcal{A}_=$-equiloaded. If the graphical representation of the automaton A does not contain any reachable directed cycle from which some accepting state is reachable, the number of accepting computation paths is finite and the implication holds trivially.

Now, let the graphical representation of the automaton A contain such directed cycle, i.e., the number of accepting computation paths be infinite. For the purpose of contradiction, let us suppose that the graphical representation of the automaton A is not strongly connected. Since the graphical representation is connected, this implies that a bridge has to exist in the graphical representation.

In other words, a transition e in D exists, such that

$$\#[e, \mathrm{Acc}(A, n)] \leq |\mathrm{Acc}(A, n)|$$

for all n in $\mathbb{N}$. However, since the number of accepting computation paths is infinite and since

$$\sum_{y \in D} \#[y, \mathrm{Acc}(A, n)] = n \cdot |\mathrm{Acc}(A, n)|,$$

a transition f in D and an infinite increasing sequence of nonnegative integers $\{n_k\}_{k=0}^{\infty}$ have to exist, such that $|\mathrm{Acc}(A, n_k)| > 0$ and

$$\#[f, \mathrm{Acc}(A, n_k)] \geq \frac{n_k}{|D|} \cdot |\mathrm{Acc}(A, n_k)|$$

for all k in $\mathbb{N}$. However, existence of such a pair of transitions clearly implies that the transition-equiloadedness $\mathcal{A}_=$-measure of the automaton A is zero.

The case $\mathcal{S} = \mathcal{C}_=$ is analogous.

Now, let us prove the more difficult right-to-left implications. First, let the graphical representation of the automaton A be without a reachable directed cycle from which some accepting state is reachable. Then, there is only a finite number of accepting computation paths. As a consequence, a nonnegative integer n_0 in $\mathbb{N}$ exists, such that there is no accepting computation path of length n, for all nonnegative integers $n \geq n_0$. Thus, both the numerator and the denominator of the n-th transition-equiloadedness $\mathcal{A}_=$-quotient are 1, for all $n \geq n_0$. This implies that $B_A(\mathcal{A}_=) = 1 > 0$, i.e., the automaton A is transition-$\mathcal{A}_=$-equiloaded, and thus also weakly transition-$\mathcal{A}_=$-equiloaded.

Further, let us suppose that the graphical representation of the automaton A does not contain any reachable directed cycle. Then a nonnegative integer n_0 in $\mathbb{N}$ exists, such that there is not any (accepting or nonaccepting) computation path of length n, for all nonnegative integers $n \geq n_0$. Thus, again, both the numerator and the denominator of the n-th transition-equiloadedness $\mathcal{C}_=$-quotient are 1, for all $n \geq n_0$. Thus, the automaton A is transition-$\mathcal{C}_=$-equiloaded, and therefore also weakly transition-$\mathcal{C}_=$-equiloaded.

Now, we shall prove that if the graphical representation of the automaton A is strongly connected, then the automaton A is both $\mathcal{C}_=$-equiloaded and $\mathcal{A}_=$-equiloaded.

Let us start with the proof of $\mathcal{C}_=$-equiloadedness. First, we shall prove that a constant M in $\mathbb{R}$ exists, such that for all nonnegative integers $n \geq |K|$, all transitions e, f in D and for all states q in K, the property

$$\#_{A_q}[f, \mathrm{Comp}(A_q, n)] \leq M \cdot \#_{A_q}[e, \mathrm{Comp}(A_q, n)] \tag{2.29}$$

holds (as above in this report, A_q denotes the automaton A with its initial state changed to q). To be more specific, we shall define the constant M as follows. Since the graphical representation of the automaton A is strongly connected, also the graphical representation of A_p is strongly connected, for each p in K. Thus, by Lemma 2.3.1,

$$|\mathrm{Comp}(A_p, n)| = \Theta(\rho^n)$$

for each p in K. Therefore, it follows that a real number $M' \geq 1$ exists, such that

$$\max_{p \in K} |\mathrm{Comp}(A_p, n)| \leq M' \cdot \min_{p \in K} |\mathrm{Comp}(A_p, n)|$$

for all n in $\mathbb{N}$. We define M by

$$M = 2|K||\Sigma|^{2|K|} \cdot M'.$$

We shall prove the inequality (2.29) by a variant of the mathematical induction. In essence, it shall be an induction on n, but in one induction step, we shall prove the property for $|K|$ values of n at once. Let the transitions e, f in D be fixed.

1. Let n be in $\{|K|, |K|+1, \ldots, 2|K|-1\}$. Since the graphical representation of the automaton A is strongly connected, it is clearly possible to reach the initial state of the transition e in at most $|K|-1$ steps from the state q. One step is needed to pass this transition and the corresponding computation path can be arbitrarily prolonged – thus, it can be prolonged also to the length of n.

   That is, we have proved that

$$\#_{A_q}[e, \mathrm{Comp}(A_q, n)] \geq 1.$$

However, on the other hand, clearly

$$\#_{A_q}[f, \mathrm{Comp}(A_q, n)] \leq 2|K||\Sigma|^{2|K|}$$

(since in each state there are at most $|\Sigma|$ transitions through which the computation path can proceed – this holds also for DFAε, since the ε-transition can lead from the state only if there is no other transition leading from that state). Thus,

$$\#_{A_q}[f, \mathrm{Comp}(A_q, n)] \leq 2|K||\Sigma|^{2|K|} \cdot \#_{A_q}[e, \mathrm{Comp}(A_q, n)].$$

That is, since $2|K||\Sigma|^{2|K|} \leq M$, the basis of the induction is proved. However, we shall keep in mind that for n in $\{|K|, |K|+1, \ldots, 2|K|-1\}$, we can use also $2|K||\Sigma|^{2|K|}$ instead of M – we shall use this fact in the proof of the induction step.

2. Let us suppose that (2.29) holds for all n in $\{|K|, |K|+1, \ldots, k|K|-1\}$. We shall prove that the property (2.29) holds also for all n in the set $\{k|K|, k|K|+1, \ldots, (k+1)|K|-1\}$.

   Let n be in $\{k|K|, k|K|+1, \ldots, (k+1)|K|-1\}$. Every computation path γ' of the automaton A_q of length n can be decomposed into a computation path γ of length $|K|$ ending in some state $p(\gamma)$ and a computation path of the automaton $A_{p(\gamma)}$ of length $n-|K|$. Thus, if we denote by $p(\gamma)$ the state, in which the computation path γ ends, we have

$$\begin{aligned}
\#_{A_q}[f, \mathrm{Comp}(A_q, n)] = & \sum_{\gamma \in \mathrm{Comp}(A_q, |K|)} \Big(\#_{A_q}[f, \gamma] \cdot |\mathrm{Comp}(A_{p(\gamma)}, n-|K|)| + \\
& + \#_{A_{p(\gamma)}}[f, \mathrm{Comp}(A_{p(\gamma)}, n-|K|)] \Big) \leq \\
\leq & \sum_{\gamma \in \mathrm{Comp}(A_q, |K|)} \Big(\#_{A_q}[f, \gamma] \cdot \max_{p \in K} |\mathrm{Comp}(A_p, n-|K|)| + \\
& + \#_{A_{p(\gamma)}}[f, \mathrm{Comp}(A_{p(\gamma)}, n-|K|)] \Big) \overset{IH}{\leq} \\
\overset{IH}{\leq} & \#_{A_q}[f, \mathrm{Comp}(A_q, |K|)] \cdot \max_{p \in K} |\mathrm{Comp}(A_p, n-|K|)| + \\
& + \sum_{\gamma \in \mathrm{Comp}(A_q, |K|)} M \cdot \#_{A_{p(\gamma)}}[e, \mathrm{Comp}(A_{p(\gamma)}, n-|K|)],
\end{aligned}$$

where IH stands for the application of the induction hypothesis. Thus, by applying the result obtained in the basis of the induction to the quantity $\#_{A_q}[f, \mathrm{Comp}(A_q, |K|)]$, we may continue the derivation as

$$\begin{aligned}
\#_{A_q}[f, \mathrm{Comp}(A_q, n)] \leq & \ 2|K||\Sigma|^{2|K|} \cdot \#_{A_q}[e, \mathrm{Comp}(A_q, |K|)] \cdot \max_{p \in K} |\mathrm{Comp}(A_p, n-|K|)| + \\
& + M \cdot \sum_{\gamma \in \mathrm{Comp}(A_q, |K|)} \#_{A_{p(\gamma)}}[e, \mathrm{Comp}(A_{p(\gamma)}, n-|K|)]
\end{aligned}$$

and since, as we have already noted,

$$\max_{p \in K} |\mathrm{Comp}(A_p, n-|K|)| \leq \frac{M}{2|K||\Sigma|^{2|K|}} \cdot \min_{p \in K} |\mathrm{Comp}(A_p, n-|K|)|,$$

we obtain

$$\begin{aligned}
\#_{A_q}[f, \mathrm{Comp}(A_q, n)] \leq & \ M \cdot \#_{A_q}[e, \mathrm{Comp}(A_q, |K|)] \cdot \min_{p \in K} |\mathrm{Comp}(A_p, n-|K|)| + \\
& + M \cdot \sum_{\gamma \in \mathrm{Comp}(A_q, |K|)} \#_{A_{p(\gamma)}}[e, \mathrm{Comp}(A_{p(\gamma)}, n-|K|)] =
\end{aligned}$$

$$
\begin{aligned}
&= M \cdot \sum_{\gamma \in \mathrm{Comp}(A_q, |K|)} \Big(\#_{A_q}[e, \gamma] \cdot \min_{p \in K} |\mathrm{Comp}(A_p, n - |K|)| + \\
&\qquad\qquad + \#_{A_{p(\gamma)}}[e, \mathrm{Comp}(A_{p(\gamma)}, n - |K|)] \Big) \leq \\
&\leq M \cdot \sum_{\gamma \in \mathrm{Comp}(A_q, |K|)} \Big(\#_{A_q}[e, \gamma] \cdot |\mathrm{Comp}(A_{p(\gamma)}, n - |K|)| + \\
&\qquad\qquad + \#_{A_{p(\gamma)}}[e, \mathrm{Comp}(A_{p(\gamma)}, n - |K|)] \Big) = \\
&= M \cdot \#_{A_q}[e, \mathrm{Comp}(A_q, n)].
\end{aligned}
$$

Thus, we have proved that the inequality (2.29) holds for all nonnegative integers $n \geq |K|$, all transitions e, f in D and for all states q in K. However, by Lemma 1.5.3,

$$
\begin{aligned}
B_A(\mathcal{C}_=) &= \min_{(e,f) \in D^2} \liminf_{n \to \infty} \frac{\#[e, \mathrm{Comp}(A, n)] + 1}{\#[f, \mathrm{Comp}(A, n)] + 1} \geq \\
&\geq \min_{(e,f) \in D^2} \liminf_{n \to \infty} \frac{\#[e, \mathrm{Comp}(A, n)] + 1}{M \cdot \#[e, \mathrm{Comp}(A, n)] + 1} = \frac{1}{M} > 0,
\end{aligned}
$$

since $\#[e, \mathrm{Comp}(A, n)] \to \infty$ for $n \to \infty$ and for all e in D. Thus, the automaton A is weakly transition-$\mathcal{C}_=$-equiloaded.

It remains to prove the implication for $\mathcal{S} = \mathcal{A}_=$. Let e, f in D be transitions. Let $\{n_k\}_{k=0}^{\infty}$ be the infinite increasing sequence of all nonnegative integers n, such that $|\mathrm{Acc}(A, n)| > 0$. First, we shall show that a constant s in $\mathbb{N}$ exists, such that for all nonnegative integers k, $n_k \geq s$, the property

$$\#[f, \mathrm{Comp}(A, n_k - s)] \leq \#[f, \mathrm{Acc}(A, n_k)] \tag{2.30}$$

holds. To prove this, let S be the set of all lengths of closed walks in the graphical representation of the automaton A, beginning and ending in the vertex corresponding to the initial state q_0. Let d be the greatest common divisor of elements of S. Clearly, the set S is closed under addition and thus, by Lemma A.5.3, contains all multiples of d greater than some nonnegative integer N in $\mathbb{N}$.

Moreover, let us define the set R as follows: a nonnegative integer n in $\mathbb{N}$ is in R, if r in $\{0, 1, \ldots, d-1\}$ exists, such that n is the smallest nonnegative integer with residue r after the division by d, such that $(q_0, w) \vdash^n (q, \varepsilon)$ for some word w in Σ^* and accepting state q in F. It is clear that R is a finite set, since it contains at most d elements. Moreover, it is clear that if n_k has a residue r after the division by d for some k in $\mathbb{N}$, then there exists an element $R(n_k)$ in R with the residue r as well. Let us denote the maximal element of R by R_{max}.

We shall prove that the inequality (2.30) holds for

$$s = N + |K| + R_{max}.$$

It clearly suffices to show that every computation path γ of length $n_k - s$ can be prolonged to at least one accepting computation path of length n_k. In fact:

1. The initial state q_0 can be reached from the resulting state of the computation path γ in at most $|K| - 1$ steps. If we denote the corresponding prolonged computation path by γ', then $|\gamma'|$ has to be divisible by d, since the walk in the graphical representation of A corresponding to the computation path γ' is obviously closed.

2. Since $n_k - R(n_k)$ and thus also $(n_k - R(n_k)) - |\gamma'|$ are divisible by d, and since the inequality $(n_k - R(n_k)) - |\gamma'| \geq N$ holds, the computation path γ' can be prolonged to a computation path γ'' of length $n_k - R(n_k)$ ending in q_0.

3. From the definition of the set R, it follows that the computation path γ'' can be prolonged to a computation path of length n_k, ending in an accepting state.

Thus, we have proved that (2.30) holds. Moreover, clearly

$$\#[f, \mathrm{Acc}(A, n_k)] \leq \#[f, \mathrm{Comp}(A, n_k)].$$

for all k in $\mathbb{N}$. Thus, for all k in $\mathbb{N}$, such that $n_k \geq s$, we have

$$\#[f, \mathrm{Comp}(A, n_k - s)] \leq \#[f, \mathrm{Acc}(A, n_k)] \leq \#[f, \mathrm{Comp}(A, n_k)]. \tag{2.31}$$

Once again, for a given computation path γ, let us denote by $p(\gamma)$ the resulting state of this computation. Now, for k in $\mathbb{N}$, such that $n_k \geq s$, we have

$$\begin{aligned}
\#[f, \mathrm{Comp}(A, n_k)] &= \sum_{\gamma \in \mathrm{Comp}(A, n_k - s)} \Big(\#[f, \gamma] \cdot |\mathrm{Comp}(A_{p(\gamma)}, s)| + \#_{A_{p(\gamma)}}[f, \mathrm{Comp}(A_{p(\gamma)}, s)] \Big) \leq \\
&\leq \#[f, \mathrm{Comp}(A, n_k - s)] \cdot \max_{p \in K} |\mathrm{Comp}(A_p, s)| + \\
&\quad + \sum_{\gamma \in \mathrm{Comp}(A, n_k - s)} \#_{A_{p(\gamma)}}[f, \mathrm{Comp}(A_{p(\gamma)}, s)] \leq \\
&\leq \#[f, \mathrm{Comp}(A, n_k - s)] \cdot \max_{p \in K} |\mathrm{Comp}(A_p, s)| + \\
&\quad + |\mathrm{Comp}(A, n_k - s)| \cdot \max_{p \in K} \#_{A_p}[f, \mathrm{Comp}(A_p, s)].
\end{aligned}$$

Now, both

$$\max_{p \in K} |\mathrm{Comp}(A_p, s)| \qquad \text{and} \qquad \max_{p \in K} \#_{A_p}[f, \mathrm{Comp}(A_p, s)]$$

are constants (they do not depend on n). Thus, if we denote by Q the greater of these two constants, we have

$$\#[f, \mathrm{Comp}(A, n_k)] \leq Q \cdot (\#[f, \mathrm{Comp}(A, n_k - s)] + |\mathrm{Comp}(A, n_k - s)|) \,.$$

However, since we have already proved that the automaton with strongly connected graphical representation is weakly transition-$\mathcal{C}_=$-equiloaded, clearly

$$\#[f, \mathrm{Comp}(A, n_k - s)] = \omega\left(|\mathrm{Comp}(A, n_k - s)|\right),$$

and thus,

$$\#[f, \mathrm{Comp}(A, n_k)] \leq 2Q \cdot \#[f, \mathrm{Comp}(A, n_k - s)] \tag{2.32}$$

for n_k greater than some N' in $\mathbb{N}$. Thus, for n_k greater than some N_0 in $\mathbb{N}$, we have

$$\begin{aligned}
\#[f, \mathrm{Acc}(A, n_k)] &\overset{(2.31)}{\leq} \#[f, \mathrm{Comp}(A, n_k)] \overset{(2.32)}{\leq} 2Q \cdot \#[f, \mathrm{Comp}(A, n_k - s)] \overset{(2.29)}{\leq} \\
&\overset{(2.29)}{\leq} 2MQ \cdot \#[e, \mathrm{Comp}(A, n_k - s)] \overset{(2.31)}{\leq} 2MQ \cdot \#[e, \mathrm{Acc}(A, n_k)],
\end{aligned}$$

since (2.31) holds also for $f = e$. Thus, by Lemma 1.5.3,

$$\begin{aligned}
B_A(\mathcal{A}_=) &= \min_{(e,f) \in D^2} \liminf_{n \to \infty} \frac{\#[e, \mathrm{Acc}(A, n)] + 1}{\#[f, \mathrm{Acc}(A, n)] + 1} \geq \\
&\geq \min_{(e,f) \in D^2} \liminf_{n \to \infty} \frac{\#[e, \mathrm{Acc}(A, n)] + 1}{2MQ \cdot \#[e, \mathrm{Acc}(A, n)] + 1} = \frac{1}{2MQ} > 0,
\end{aligned}$$

i.e., the automaton A is weakly transition-$\mathcal{A}_=$-equiloaded. The theorem is proved. □

In the theorem that follows, we shall prove the characterization of weakly transition-$\mathcal{C}_\leq$-equiloaded and weakly transition-$\mathcal{A}_\leq$-equiloaded deterministic finite automata. We shall prove that for DFA and DFAε, the weak transition-$\mathcal{C}_\leq$-equiloadedness is equivalent to the weak transition-$\mathcal{C}_=$-equiloadedness, and that the weak transition-$\mathcal{A}_\leq$-equiloadedness is equivalent to the weak transition-$\mathcal{A}_=$-equiloadedness.

Theorem 2.3.19 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε with connected graphical representation.

a) A is weakly transition-$\mathcal{C}_\leq$-equiloaded, if and only if its graphical representation either does not contain any reachable directed cycle, or is strongly connected. That is, the automaton A is weakly transition-$\mathcal{C}_\leq$-equiloaded, if and only if the automaton A is weakly transition-$\mathcal{C}_=$-equiloaded.

b) A is weakly transition-$\mathcal{A}_\leq$-equiloaded, if and only if its graphical representation either does not contain any reachable directed cycle from which some accepting state is reachable, or is strongly connected. That is, the automaton A is weakly transition-$\mathcal{A}_\leq$-equiloaded, if and only if the automaton A is weakly transition-$\mathcal{A}_=$-equiloaded.

Proof. First, let us suppose that the graphical representation of the automaton A does not contain any reachable directed cycle from which some accepting state is reachable. Then, there is only a finite number of accepting computation paths of the automaton A, and thus, a nonnegative integer n_0 in $\mathbb{N}$ exists, such that for all $n \geq n_0$ and for each transition e in D, the property

$$\#[e, \mathrm{Acc}(A, \leq n)] = \#[e, \mathrm{Acc}(A, \leq n_0)]$$

holds. Thus,

$$B_A(\mathcal{A}_\leq) = \min_{(e,f)\in D^2} \frac{\#[e, \mathrm{Acc}(A, \leq n_0)] + 1}{\#[f, \mathrm{Acc}(A, \leq n_0)] + 1} > 0,$$

i.e., the automaton A is weakly transition-$\mathcal{A}_\leq$-equiloaded.

Similarly, let us suppose that the graphical representation of the automaton A does not contain any reachable directed cycle. Then, by the same reasoning as above,

$$B_A(\mathcal{C}_\leq) = \min_{(e,f)\in D^2} \frac{\#[e, \mathrm{Comp}(A, \leq n_0)] + 1}{\#[f, \mathrm{Comp}(A, \leq n_0)] + 1} > 0,$$

and the automaton A is weakly transition-$\mathcal{C}_\leq$-equiloaded.

Now, let us suppose that the automaton A contains at least one reachable directed cycle from which some accepting state is reachable. Then, by Theorem 2.3.16, the equality

$$B_A(\mathcal{A}_\leq) = B_A(\mathcal{A}_=)$$

holds. Thus, by Theorem 2.3.18, it follows that the automaton A is weakly transition-$\mathcal{A}_\leq$-equiloaded, if and only if its graphical representation is strongly connected.

Finally, let us suppose that the automaton A contains at least one reachable directed cycle. Then, for each transition e in D and all n in $\mathbb{N}$, the property

$$\#[e, \mathrm{Comp}(A, \leq n)] = \#[e, \mathrm{Acc}(A^K, \leq n)]$$

holds, where $A^K = (K, \Sigma, \delta, q_0, K)$ is the automaton A with all states accepting. Thus, by Theorem 2.3.16, we obtain

$$B_A(\mathcal{C}_\leq) = B_{A^K}(\mathcal{A}_\leq) = B_{A^K}(\mathcal{A}_=).$$

Since the graphical representation of the automaton A is strongly connected if and only if the graphical representation of the automaton A^K is strongly connected, it follows from Theorem 2.3.18 that the automaton A is weakly transition-$\mathcal{C}_\leq$-equiloaded, if and only if its graphical representation is strongly connected. Thus, the theorem is proved. □

Before we turn our attention to weakly state-$\mathcal{S}$-equiloaded DFA and DFAε, we shall state a theorem that we have proved in [25] for weakly transition-$\mathcal{A}_=$-equiloaded DFA (without ε-transitions), and that can be clearly extended to hold also for another choices of $\mathcal{S}$. Since the original proof is not very complicated, we shall not present it here. We shall prove only the extension of the theorem to $\mathcal{S}$ in $\{\mathcal{C}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$.

Theorem 2.3.20 Let L in $\mathscr{R}$ be a regular language. Let $\mathcal{S}$ be in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$. The language L is a weakly transition-$\mathcal{S}$-equiloaded DFA-language, if and only if the minimal DFA accepting L is weakly transition-$\mathcal{S}$-equiloaded.

Proof. The proof for $\mathcal{S} = \mathcal{A}_=$ can be found in [25]. By Theorem 2.3.19, the theorem holds also for $\mathcal{S} = \mathcal{A}_\leq$.

Now, by Theorem 2.3.18, it is clear that $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{C}_=) \subseteq \mathscr{L}_{\delta-WEQ-DFA}(\mathcal{A}_=)$ (in fact, we shall prove in Subsection 2.3.6 that these families are equal). That is, if L is in $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{C}_=)$, then it is also in $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{A}_=)$ and thus, the minimal DFA accepting L is weakly transition-$\mathcal{A}_=$-equiloaded. Moreover, it is clear that if the graphical representation of the minimal DFA accepting L does not contain any reachable directed cycle from which some accepting state is reachable, then it also does not contain any other reachable directed cycle – otherwise, it would be possible to delete the states of the cycle without changing the accepted language, and that would be a contradiction with the assumption that the given automaton is minimal. However, this property together with Theorem 2.3.18 implies that this minimal DFA is also weakly transition-$\mathcal{C}_=$-equiloaded. Since the converse implication is trivial, the theorem holds also for the case $\mathcal{S} = \mathcal{C}_=$.

The last remaining case, $\mathcal{S} = \mathcal{C}_\leq$, follows directly from the case $\mathcal{S} = \mathcal{C}_=$ and from Theorem 2.3.19. □

In what follows, we shall characterize the family of weakly state-$\mathcal{C}_=$-equiloaded deterministic finite automata. However, before we state this characterization (Theorem 2.3.31), we shall introduce some notation and prove several technical lemmas that we shall use in its proof.

Notation 2.3.21 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε, such that its graphical representation has k strongly connected components, corresponding to disjoint sets of states $K_0, K_1, \ldots, K_{k-1}$, such that

$$\bigcup_{i=0}^{k-1} K_i = K,$$

and q_0 is in K_0. Let i be in $\{0, 1, \ldots, k-1\}$, and q be in K_i. Then, by $A[K_i, q]$, we denote the automaton obtained by the restriction of the set of states of A to K_i, and by choosing q as an initial state of the resulting automaton. Formally, $A[K_i, q] = (K_i, \Sigma, \delta', q, F \cap K_i)$, where, for p, q in K_i and c in Σ, $\delta'(p, c) = q$ whenever $\delta(p, c) = q$.

Definition 2.3.22 Let the symbols have the same meaning as in Notation 2.3.21. We shall define the *SCC-dag* $SCC(A)$ of the automaton A as follows: $SCC(A)$ is a directed acyclic graph with the set of vertices $\{K_0, K_1, \ldots, K_{k-1}\}$, and with an edge from the vertex K_i to the vertex K_j (for some i, j in $\{0, 1, \ldots, k-1\}$), if and only if states p in K_i and q in K_j exist, such that $(p, c) \vdash (q, \varepsilon)$ for some c in $\Sigma \cup \{\varepsilon\}$.

Notation 2.3.23 Let the symbols have the same meaning as above. Let $x = (K_{i_1}, K_{i_2}, \ldots, K_{i_s})$ be a path in $SCC(A)$. Let q be a state in K_{i_1}. Then, by $\mathrm{Comp}_x(A_q, n)$, we denote the set of computation paths γ of the automaton A_q, such that γ follows the path x, i.e., it first visits some nonzero number of states in K_{i_1}, then some nonzero number of states in K_{i_2}, etc., and finally ends in some state in K_{i_s}. Similarly, by $\mathrm{Acc}_x(A_q, n)$, we denote the set of all such accepting computation paths.

Notation 2.3.24 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε, let K_i and K_j be vertices of its SCC-dag. Then, by $\mathrm{Pth}[K_i, K_j]$, we denote the set of all paths in $SCC(A)$, from the vertex K_i to the vertex K_j.

Lemma 2.3.25 Let the symbols have the same meaning as above. Let $x = (K_{i_1}, K_{i_2}, \ldots, K_{i_s})$ be a path in $SCC(A)$. Let q be a state in K_{i_1}. For $j = 0, 1, \ldots, k-1$, let us denote by ρ_j the Perron-Frobenius eigenvalue of the transition matrix $\Delta_{A[K_j, q_j]}$ of the automaton $A[K_j, q_j]$, where q_j is an arbitrary[11] state in K_j, for $j = 1, \ldots, k-1$. Let us denote by ρ_{max} the greatest of the Perron-Frobenius eigenvalues corresponding to strongly connected components on the path x, i.e.,

$$\rho_{max} = \max_{j=1,\ldots,s} \rho_{i_j}.$$

Then, if $\rho_{max} > 0$, the number of computation paths in $\mathrm{Comp}_x(A_q, n)$ is asymptotically equal to

$$|\mathrm{Comp}_x(A_q, n)| = \Theta\left(n^{r-1} \cdot \rho_{max}^n\right),$$

where r is the number of strongly connected components corresponding to vertices on x, such that their Perron-Frobenius eigenvalue is ρ_{max}, i.e.,

$$r = \left|\left\{ j \in \{1, \ldots, s\} \;\middle|\; \rho_{i_j} = \rho_{max} \right\}\right|.$$

If $\rho_{max} = 0$, then a nonnegative integer constant R exists, such that

$$|\mathrm{Comp}_x(A_q, n)| = R \cdot [n = s].$$

Proof. For $j = 1, \ldots, s$, let us denote by x_j the path $x_j = (K_{i_1}, K_{i_2}, \ldots, K_{i_j})$. We shall prove by induction on j that the statement of the lemma holds for all x_j, $j = 1, \ldots, s$.

For this purpose, let us denote by $\rho_{max}^{(j)}$ the greatest of the Perron-Frobenius eigenvalues corresponding to strongly connected components on the path x_j, and by $r^{(j)}$, the number of strongly connected components corresponding to vertices on x_j, such that their Perron-Frobenius eigenvalue is $\rho_{max}^{(j)}$. That is, $\rho_{max}^{(j)}$ and $r^{(j)}$ are defined analogously as ρ_{max} and r, but for the path x_j instead of x.

1. For $j = 1$, it is obvious that the property

$$|\mathrm{Comp}_{x_1}(A_q, n)| = |\mathrm{Comp}(A[K_{i_1}, q], n)|$$

holds. However, the graphical representation of the automaton $A[K_{i_1}, q]$ is strongly connected, and thus, by Lemma 2.3.1,

$$|\mathrm{Comp}(A[K_{i_1}, q], n)| = \Theta(\rho_{i_1}^n).$$

However, since K_{i_1} is the only vertex on the path x_1, clearly $\rho_{max}^{(1)} = \rho_{i_1}$, and $r^{(1)} = 1$. Thus, the basis of the induction is proved.

2. Let us suppose that the statement of the lemma holds for all x_j, $j = 1, \ldots, t$, for some t in $\{1, \ldots, s-1\}$. We shall prove that it holds also for x_{t+1}. Let q_{trans} in K_{i_t} be a state, such that a state q'_{trans} in $K_{i_{t+1}}$ exists, such that

$$(q_{trans}, c) \vdash (q'_{trans}, \varepsilon)$$

for some c in $\Sigma \cup \{\varepsilon\}$. Let $P = P_{A[K_{i_t}, q_t]}$ be the period of the automaton $A[K_{i_t}, q_t]$. Then, obviously, at least one number p in $\{0, 1, \ldots, P-1\}$ and a nonnegative integer n_0 in $\mathbb{N}$ exists, such that

$$\mathrm{Acc}_{x_t}(A_q^{\{q_{trans}\}}, n \cdot P + p)$$

is nonempty for all $n \geq n_0$. Thus, it can be easily seen (a similar reasoning has been used a number of times up to now in this report) that a constant C in $\mathbb{N}$ exists, such that for $n \geq n_0$,

[11] This eigenvalue is obviously the same, no matter what state q_j in K_j is chosen.

every computation path of length $(n-C)\cdot P+p$ can be prolonged to a computation path of length $n\cdot P+p$, ending in q_{trans}. In other words,

$$\begin{aligned}\left|\mathrm{Acc}_{x_t}(A_q^{\{q_{trans}\}}, n\cdot P+p)\right| \geq \left|\mathrm{Comp}_{x_t}(A_q,(n-C)\cdot P+p)\right| \geq \\ \geq C'\cdot\left|\mathrm{Comp}_{x_t}(A_q, n\cdot P+p)\right|,\end{aligned} \tag{2.33}$$

for some positive real constant C' in $\mathbb{R}^+$ (the second inequality is a direct consequence of the induction hypothesis).

Now, it is obvious that the number of computation paths in $\mathrm{Comp}_{x_{t+1}}(A_q,n)$ is greater than or equal to the number of uses of the transition (q_{trans},c,q'_{trans}) in these computation paths. Moreover, by a direct combinatorial insight, it is clear that Lemma 2.3.6 can be generalized to hold also for $\mathrm{Comp}_{x_{t+1}}(A_q,n)$ instead of $\mathrm{Comp}(A,n)$. That is,

$$\begin{aligned}\left|\mathrm{Comp}_{x_{t+1}}(A_q,n)\right| \geq \#\left[(q_{trans},c,q'_{trans}),\mathrm{Comp}_{x_{t+1}}(A_q,n)\right] = \\ = \sum_{i=0}^{n-1}\left|\mathrm{Acc}_{x_t}(A_q^{\{q_{trans}\}},i)\right|\cdot\left|\mathrm{Comp}(A[K_{i_{t+1}},q'_{trans}],n-i-1)\right|.\end{aligned} \tag{2.34}$$

Further, from the induction hypothesis and Lemma 2.3.1, it is clear that a positive real constant D in $\mathbb{R}^+$ exists, such that

$$\begin{aligned}\left|\mathrm{Comp}_{x_t}(A_q,i\cdot P+p)\right|\cdot\left|\mathrm{Comp}(A[K_{i_{t+1}},q'_{trans}],n-(i\cdot P+p)-1)\right| \geq \\ \geq D\cdot\sum_{l=0}^{2P-1}\left|\mathrm{Comp}_{x_t}(A_q,i\cdot P+l)\right|\cdot\left|\mathrm{Comp}(A[K_{i_{t+1}},q'_{trans}],n-(i\cdot P+l)-1)\right|.\end{aligned} \tag{2.35}$$

Thus, with use of inequalities (2.33) and (2.35), we may continue[12] the derivation (2.34) as follows:

$$\begin{aligned}\left|\mathrm{Comp}_{x_{t+1}}(A_q,n)\right| &\geq \sum_{i=0}^{n-1}\left|\mathrm{Acc}_{x_t}(A_q^{\{q_{trans}\}},i)\right|\cdot\left|\mathrm{Comp}(A[K_{i_{t+1}},q'_{trans}],n-i-1)\right| \geq \\ &\geq \sum_{i=0}^{\lfloor(n-1-p)/P\rfloor}\left|\mathrm{Acc}_{x_t}(A_q^{\{q_{trans}\}},i\cdot P+p)\right|\cdot \\ &\qquad\cdot\left|\mathrm{Comp}(A[K_{i_{t+1}},q'_{trans}],n-(i\cdot P+p)-1)\right| \overset{(2.33)}{\geq} \\ &\overset{(2.33)}{\geq} C'\cdot\sum_{i=0}^{\lfloor(n-1-p)/P\rfloor}\left|\mathrm{Comp}_{x_t}(A_q^{\{q_{trans}\}},i\cdot P+p)\right|\cdot \\ &\qquad\cdot\left|\mathrm{Comp}(A[K_{i_{t+1}},q'_{trans}],n-(i\cdot P+p)-1)\right| \overset{(2.35)}{\geq} \\ &\overset{(2.35)}{\geq} C'\cdot D\cdot\sum_{i=0}^{n-1}\left|\mathrm{Comp}_{x_t}(A_q,i)\right|\cdot\left|\mathrm{Comp}(A[K_{i_{t+1}},q'_{trans}],n-i-1)\right|.\end{aligned}$$

Now, let us first examine the case $0<\rho_{i_{t+1}}=\rho_{max}^{(t)}=\rho_{max}^{(t+1)}=:\rho$. Then, $r^{(t+1)}=r^{(t)}+1$, and by the use of induction hypothesis and Lemma 2.3.1, we obtain

$$\begin{aligned}\left|\mathrm{Comp}_{x_{t+1}}(A_q,n)\right| \geq C'\cdot D\cdot\sum_{i=0}^{n-1}\Theta\left(i^{r^{(t)}-1}\cdot\rho^i\right)\cdot\Theta\left(\rho^{n-i-1}\right) = \\ = E\cdot\rho^n\sum_{i=0}^{n-1}i^{r^{(t)}-1} = \Theta\left(n^{r^{(t+1)}-1}\cdot\rho^n\right).\end{aligned}$$

[12] At least for automata with infinite number of accepting computation paths in $\mathrm{Acc}_{x_t}(A_q^{\{q_{trans}\}})$. However, the case of automata with finite number of such computation paths is trivial.

where E in $\mathbb{R}^+$ is a positive real constant. Thus, in this case,

$$\left|\mathrm{Comp}_{x_{t+1}}(A_q, n)\right| = \Omega\left(n^{r^{(t+1)}-1} \cdot \rho^n\right).$$

We consider to be obvious that the theorem holds also for the case $0 = \rho_{i_{t+1}} = \rho_{max}^{(t)} = \rho_{max}^{(t+1)}$.

Now, let us examine the case $\rho_{i_{t+1}} = \rho_{max}^{(t+1)} > \rho_{max}^{(t)} > 0$. Then, $r^{(t+1)} = 1$. The reader may easily convince himself that for arbitrary two distinct real numbers, a, b in $\mathbb{R}$, $a > b$, the formula

$$\sum_{i=0}^{n} a^i b^{n-i-1} = \frac{-a}{b-a} \cdot a^n + \frac{b}{b-a} \cdot b^n = \Theta(a^n) \tag{2.36}$$

holds. We obtain

$$\left|\mathrm{Comp}_{x_{t+1}}(A_q, n)\right| \geq C' \cdot D \cdot \sum_{i=0}^{n-1} \Theta\left(i^{r^{(t)}-1} \cdot \left(\rho_{max}^{(t)}\right)^i\right) \cdot \Theta\left(\rho_{i_{t+1}}^{n-i-1}\right).$$

For some $\varepsilon > 0$, such that $\rho_{max}^{(t)} + \varepsilon < \rho_{i_{t+1}}$, we have

$$\begin{aligned}
&\sum_{i=0}^{n-1} \Theta\left(\left(\rho_{max}^{(t)}\right)^i\right) \cdot \Theta\left(\rho_{i_{t+1}}^{n-i-1}\right) \preceq \\
\preceq &\sum_{i=0}^{n-1} \Theta\left(i^{r^{(t)}-1} \cdot \left(\rho_{max}^{(t)}\right)^i\right) \cdot \Theta\left(\rho_{i_{t+1}}^{n-i-1}\right) \preceq \\
\preceq &\sum_{i=0}^{n-1} \Theta\left(\left(\rho_{max}^{(t)} + \varepsilon\right)^i\right) \cdot \Theta\left(\rho_{i_{t+1}}^{n-i-1}\right).
\end{aligned}$$

However, both

$$\sum_{i=0}^{n-1} \Theta\left(\left(\rho_{max}^{(t)}\right)^i\right) \cdot \Theta\left(\rho_{i_{t+1}}^{n-i-1}\right),$$

and

$$\sum_{i=0}^{n-1} \Theta\left(\left(\rho_{max}^{(t)} + \varepsilon\right)^i\right) \cdot \Theta\left(\rho_{i_{t+1}}^{n-i-1}\right)$$

are $\Theta\left(\rho_{i_{t+1}}^n\right)$, by (2.36). Thus, also

$$\sum_{i=0}^{n-1} \Theta\left(i^{r^{(t)}-1} \cdot \left(\rho_{max}^{(t)}\right)^i\right) \cdot \Theta\left(\rho_{i_{t+1}}^{n-i-1}\right) = \Theta\left(\rho_{i_{t+1}}^n\right),$$

and that implies

$$\left|\mathrm{Comp}_{x_{t+1}}(A_q, n)\right| = \Omega\left(n^{r^{(t+1)}-1} \cdot \left(\rho_{max}^{(t+1)}\right)^n\right).$$

The proof of the previous case with $\rho_{max}^{(t)} = 0$ is left to the reader.

Finally, let us examine the case $0 < \rho_{i_{t+1}} < \rho_{max}^{(t+1)} = \rho_{max}^{(t)}$. In this case, we have $r^{(t+1)} = r^{(t)}$. By a slightly more involved process than in the previous case, it is possible to prove that

$$\sum_{i=0}^{n-1} \Theta\left(i^{r^{(t)}-1} \cdot \left(\rho_{max}^{(t)}\right)^i\right) \cdot \Theta\left(\rho_{i_{t+1}}^{n-i-1}\right) = \Theta\left(n^{r^{(t)}-1} \cdot \left(\rho_{max}^{(t)}\right)^n\right),$$

and thus also

$$\left|\mathrm{Comp}_{x_{t+1}}(A_q, n)\right| = \Omega\left(n^{r^{(t+1)}-1} \cdot \left(\rho_{max}^{(t+1)}\right)^n\right).$$

The proof of the previous case with $\rho_{i_{t+1}} = 0$ is left to the reader, once again.

Thus, we have proved that

$$\left|\mathrm{Comp}_{x_{t+1}}(A_q, n)\right| = \Omega\left(n^{r^{(t+1)}-1} \cdot \left(\rho_{max}^{(t+1)}\right)^n\right)$$

holds for all three possible cases, and it remains to prove that

$$\left|\mathrm{Comp}_{x_{t+1}}(A_q, n)\right| = O\left(n^{r^{(t+1)}-1} \cdot \left(\rho_{max}^{(t+1)}\right)^n\right).$$

However, this asymptotic relation may be proved in a similar manner (this estimate is easier) and the details are left to the reader.

The lemma is proved. $\square$

Lemma 2.3.26 Let the symbols have the same meaning as in Notation 2.3.21. Let q be a state in K_j, for some j in $\{0, 1, \ldots, k-1\}$. Let s be in $\{0, 1, \ldots, k-1\}$. Then,

$$\left|\mathrm{Acc}(A_q^{K_s}, n)\right| = \sum_{x \in \mathrm{Pth}[K_j, K_s]} \left|\mathrm{Comp}_x(A_q, n)\right| = \Theta\left(\max_{x \in \mathrm{Pth}[K_j, K_s]} \left|\mathrm{Comp}_x(A_q, n)\right|\right).$$

Proof. It is clear that every computation path γ in $\mathrm{Acc}(A_q^{K_s}, n)$ is also in $\mathrm{Comp}_x(A_q, n)$, for some x in $\mathrm{Pth}[K_j, K_s]$. The converse is also obviously true. Moreover, if γ_1 is in $\mathrm{Comp}_x(A_q, n)$, and γ_2 is in $\mathrm{Comp}_y(A_q, n)$, for some x, y in $\mathrm{Pth}[K_j, K_s]$, $x \neq y$, then clearly $\gamma_1 \neq \gamma_2$. That is, the first identity is proved.

Now, obviously,

$$\max_{x \in \mathrm{Pth}[K_j, K_s]} \left|\mathrm{Comp}_x(A_q, n)\right| \geq \frac{1}{|\mathrm{Pth}[K_j, K_s]|} \cdot \sum_{x \in \mathrm{Pth}[K_j, K_s]} \left|\mathrm{Comp}_x(A_q, n)\right|,$$

and

$$\max_{x \in \mathrm{Pth}[K_j, K_s]} \left|\mathrm{Comp}_x(A_q, n)\right| \leq \sum_{x \in \mathrm{Pth}[K_j, K_s]} \left|\mathrm{Comp}_x(A_q, n)\right|,$$

and since $1/|\mathrm{Pth}[K_j, K_s]|$ is a constant, the second asymptotic identity is proved as well. Thus, the lemma is proved. $\square$

Notation 2.3.27 Let the symbols have the same meaning as in Notation 2.3.21. Let K_i, K_j, K_l be vertices of $SCC(A)$. Then, by $\mathrm{Pth}[K_i, K_j, K_l]$, we shall denote the set of all paths $x = (K_{i_1}, K_{i_2}, \ldots, K_{i_s})$ in $SCC(A)$, such that r in $\{1, 2, \ldots, s\}$ exists, such that $(K_{i_1}, \ldots, K_{i_r})$ is in $\mathrm{Pth}[K_i, K_j]$, and $(K_{i_r}, \ldots, K_{i_s})$ is in $\mathrm{Pth}[K_j, K_l]$. That is, by $\mathrm{Pth}[K_i, K_j, K_l]$ we denote the set of all paths x in $SCC(A)$ from the vertex K_i to the vertex K_l, such that the vertex K_j lies on x.

Furthermore, we shall use a notation

$$\mathrm{Pth}[K_i, K_j, *] = \bigcup_{l=0}^{k-1} \mathrm{Pth}[K_i, K_j, K_l].$$

Lemma 2.3.28 Let the symbols have the same meaning as in Notation 2.3.21. Let q be a state in K_j, for some j in $\{0, 1, \ldots, k-1\}$. Let s be a member of the set $\{0, 1, \ldots, k-1\}$. Let us denote by ρ_j the Perron-Frobenius eigenvalue corresponding to K_j, and by ρ_{max} the greatest of the Perron-Frobenius eigenvalues corresponding to strongly connected components on paths in $\mathrm{Pth}[K_0, K_j, K_s]$. Let us denote by r the greatest positive integer, such that a path x in $\mathrm{Pth}[K_0, K_j, K_s]$ exists, such that there are r strongly connected components on x with the Perron-Frobenius eigenvalue equal to ρ_{max}. Then:

(i) If $\rho_{max} > 0$, and $\rho_j = \rho_{max}$, then

$$\#\left[q, \mathrm{Acc}(A^{K_s}, n)\right] = \Theta\left(n^r \cdot \rho_{max}^n\right).$$

(ii) If $\rho_{max} > 0$, and $\rho_j < \rho_{max}$, then

$$\#\left[q, \mathrm{Acc}(A^{K_s}, n)\right] = \Theta\left(n^{r-1} \cdot \rho_{max}^n\right).$$

(iii) If $\rho_{max} = \rho_j = 0$, then

$$\#\left[q, \mathrm{Acc}(A^{K_s}, n)\right] = 0$$

for all n greater than some n_0 in $\mathbb{N}$.

Proof. We shall prove these three claims separately:

(i) In this case, there has to be a path x_1 in $\mathrm{Pth}[K_0, K_j]$ with r_1 strongly connected components with Perron-Frobenius eigenvalue ρ_{max}, and a path x_2 in $\mathrm{Pth}[K_j, K_s]$ with r_2 strongly connected components with Perron-Frobenius eigenvalue ρ_{max}, so that $r_1 + r_2 = r + 1$. Moreover, since K_j lies both on x_1 and x_2, it follows that $r_1 > 0$, and also $r_2 > 0$.

Thus, from Lemma 2.3.25 and Lemma 2.3.26, it follows that

$$\left|\mathrm{Acc}(A^{K_j}, n)\right| = \Theta\left(n^{r_1-1} \cdot \rho_{max}^n\right).$$

Similarly,

$$\left|\mathrm{Acc}(A_q^{K_s}, n)\right| = \Theta\left(n^{r_2-1} \cdot \rho_{max}^n\right).$$

Moreover, by a standard reasoning concerning the period of the j-th strongly connected component, it is possible to prove that

$$\sum_{i=0}^{n} \left|\mathrm{Acc}(A^{\{q\}}, i)\right| \cdot \left|\mathrm{Acc}(A_q^{K_s}, n-i)\right| = \Theta\left(\sum_{i=0}^{n} \left|\mathrm{Acc}(A^{K_j}, i)\right| \cdot \left|\mathrm{Acc}(A_q^{K_s}, n-i)\right|\right)$$

(the definition of Θ that is required to hold for all n greater than some n_0 in $\mathbb{N}$ is used here). Thus, by Lemma 2.3.7, we have

$$\begin{aligned}
\#\left[q, \mathrm{Acc}(A^{K_s}, n)\right] &= \sum_{i=0}^{n} \left|\mathrm{Acc}(A^{\{q\}}, i)\right| \cdot \left|\mathrm{Acc}(A_q^{K_s}, n-i)\right| = \\
&= \Theta\left(\sum_{i=0}^{n} \left|\mathrm{Acc}(A^{K_j}, i)\right| \cdot \left|\mathrm{Acc}(A_q^{K_s}, n-i)\right|\right) \\
&= \sum_{i=0}^{n} \Theta\left(i^{r_1-1} \cdot \rho_{max}^i\right) \cdot \Theta\left((n-i)^{r_2-1} \cdot \rho_{max}^{n-i}\right) = \\
&= \Theta\left(\rho_{max}^n \cdot \sum_{i=0}^{n} i^{r_1-1} \cdot (n-i)^{r_2-1}\right) = \Theta\left(n^{r_1+r_2-1} \cdot \rho_{max}^n\right) = \\
&= \Theta\left(n^r \cdot \rho_{max}^n\right).
\end{aligned}$$

(ii) In this case, there has to be a path x_1 in $\mathrm{Pth}[K_0, K_j]$ with r_1 strongly connected components with Perron-Frobenius eigenvalue ρ_{max}, and a path x_2 in $\mathrm{Pth}[K_j, K_s]$ with r_2 strongly connected components with Perron-Frobenius eigenvalue ρ_{max}, so that $r_1 + r_2 = r$.

Then, the result may be proved in a similar manner that in the case of claim (i), with the difference that the case $r_1 = 0$ (resp. $r_2 = 0$) has to be handled as well.

(iii) This claim is considered to be obvious.

The lemma is proved. □

Lemma 2.3.29 Let the symbols have the same meaning as in Notation 2.3.21. Let q be a state in K_j, for some j in $\{0, 1, \ldots, k-1\}$. Let us denote by ρ_j the Perron-Frobenius eigenvalue corresponding to K_j, and by ρ_{max} the greatest of the Perron-Frobenius eigenvalues corresponding to strongly connected components on paths in $\mathrm{Pth}[K_0, K_j, *]$. Let us denote by r the greatest positive integer, such that a path x in $\mathrm{Pth}[K_0, K_j, *]$ exists, such that there are r strongly connected components on x with the Perron-Frobenius eigenvalue equal to ρ_{max}. Then:

(i) If $\rho_{max} > 0$, and $\rho_j = \rho_{max}$, then
$$\#\left[q, \mathrm{Comp}(A, n)\right] = \Theta\left(n^r \cdot \rho_{max}^n\right).$$

(ii) If $\rho_{max} > 0$, and $\rho_j < \rho_{max}$, then
$$\#\left[q, \mathrm{Comp}(A, n)\right] = \Theta\left(n^{r-1} \cdot \rho_{max}^n\right).$$

(iii) If $\rho_{max} = \rho_j = 0$, then
$$\#\left[q, \mathrm{Comp}(A, n)\right] = 0$$
for all n greater than some n_0 in $\mathbb{N}$.

Proof. The lemma is a direct consequence of Lemma 2.3.28, and of the obvious fact that
$$\#\left[q, \mathrm{Comp}(A, n)\right] = \sum_{i=0}^{k-1} \#\left[q, \mathrm{Acc}(A^{K_i}, n)\right],$$
where k is a constant. □

Example 2.3.30 Now, we shall demonstrate Lemma 2.3.29 on an example. Let us consider a deterministic finite automaton $A = (K, \Sigma, \delta, q_0, F)$ defined as follows: $K = \{q_0, q_1, \ldots, q_9\}$, $\Sigma = \{a, b, c, d\}$, and

$$\begin{array}{llll}
\delta(q_0, a) = q_1, & \delta(q_0, b) = q_1, & \delta(q_1, a) = q_0, & \delta(q_1, b) = q_0, \\
\delta(q_1, c) = q_2, & \delta(q_1, d) = q_4, & \delta(q_2, a) = q_3, & \delta(q_2, b) = q_3, \\
\delta(q_3, a) = q_2, & \delta(q_3, b) = q_2, & \delta(q_4, a) = q_5, & \delta(q_5, a) = q_4, \\
\delta(q_5, c) = q_6, & \delta(q_6, a) = q_7, & \delta(q_6, b) = q_7, & \delta(q_7, a) = q_6, \\
\delta(q_7, b) = q_6, & \delta(q_7, c) = q_8, & \delta(q_8, a) = q_9, & \delta(q_8, b) = q_9, \\
\delta(q_9, a) = q_8, & \delta(q_9, b) = q_8. & &
\end{array}$$

The set of accepting states F does not matter for the purposes of this example, let us for instance suppose that $F = K$. The graphical representation of the automaton A is depicted in Figure 2.4 (without irrelevant character labels).

The graphical representation of the automaton A has 5 strongly connected components, corresponding to the sets of states

$$\begin{aligned}
K_0 &= \{q_0, q_1\}, \\
K_1 &= \{q_2, q_3\}, \\
K_2 &= \{q_4, q_5\}, \\
K_3 &= \{q_6, q_7\}, \\
K_4 &= \{q_8, q_9\}.
\end{aligned}$$

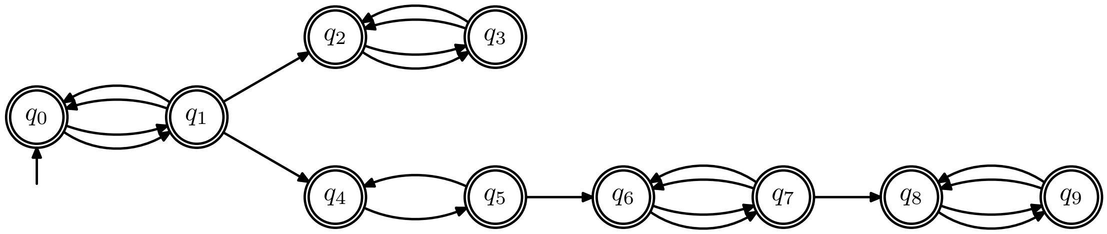


Figure 2.4: The automaton A (the character labels at transitions are not depicted).

The corresponding Perron-Frobenius eigenvalues are $\rho_0 = \rho_1 = \rho_3 = \rho_4 = 2$, and $\rho_2 = 1$. Let us examine the number of uses of states q_0, q_2, q_4, and q_6. By Lemma 2.3.29, it follows that

$$\begin{aligned}
\#\left[q_0, \mathrm{Comp}(A, n)\right] &= \Theta(n^3 \cdot 2^n), \\
\#\left[q_2, \mathrm{Comp}(A, n)\right] &= \Theta(n^2 \cdot 2^n), \\
\#\left[q_4, \mathrm{Comp}(A, n)\right] &= \Theta(n^2 \cdot 2^n), \\
\#\left[q_6, \mathrm{Comp}(A, n)\right] &= \Theta(n^3 \cdot 2^n)
\end{aligned}$$

(to the states q_0, q_2, and q_6, the point (i) of Lemma 2.3.29 applies, to the state q_4, the point (ii) of the lemma applies).

In deed, by numerically computing these quantities by the method presented in Section 2.1, we may obtain the approximate[13] results

$$\begin{aligned}
\#\left[q_0, \mathrm{Comp}(A, n)\right] \approx\ & 0.0017 \cdot n^3 \cdot 2^n + 0.0668 \cdot n^2 \cdot 2^n + 0.7781 \cdot n \cdot 2^n + 0.5111 \cdot 2^n + \\
& + 0.0026 \cdot n^2 \cdot (-2)^n + 0.1502 \cdot n \cdot (-2)^n + 0.2914 \cdot (-2)^n + \\
& + 0.1852 + 0.0123 \cdot (-1)^n, \\
\#\left[q_2, \mathrm{Comp}(A, n)\right] \approx\ & 0.0625 \cdot n^2 \cdot 2^n + 0.0625 \cdot n \cdot 2^n - 0.0312 \cdot 2^n + 0.0625 \cdot n \cdot (-2)^n + \\
& + 0.0312 \cdot (-2)^n, \\
\#\left[q_4, \mathrm{Comp}(A, n)\right] \approx\ & 0.0139 \cdot n^2 \cdot 2^n - 0.0231 \cdot n \cdot 2^n + 0.6042 \cdot 2^n + 0.0139 \cdot n \cdot (-2)^n + \\
& + 0.2106 \cdot (-2)^n - 0.2778 \cdot n - 0.6481 - 0.0185 \cdot n \cdot (-1)^n - \\
& - 0.1667 \cdot (-1)^n, \\
\#\left[q_6, \mathrm{Comp}(A, n)\right] \approx\ & 0.0017 \cdot n^3 \cdot 2^n + 0.0043 \cdot n^2 \cdot 2^n - 0.0344 \cdot n \cdot 2^n + 0.0423 \cdot 2^n + \\
& + 0.0026 \cdot n^2 \cdot (-2)^n + 0.0043 \cdot n \cdot (-2)^n - 0.0177 \cdot (-2)^n - 0.0370 + \\
& + 0.0123 \cdot (-1)^n.
\end{aligned}$$

The above derived asymptotic estimates are clearly satisfied.

Now, we shall finally use the theory developed to prove the characterization of the family of weakly state-$\mathcal{C}_=$-equiloaded deterministic finite automata.

Theorem 2.3.31 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFAε having a connected graphical representation, such that its graphical representation has k strongly connected components, corresponding to disjoint sets of states $K_0, K_1, \ldots, K_{k-1}$, such that

$$\bigcup_{i=0}^{k-1} K_i = K,$$

[13]The coefficients have been computed numerically, and thus, they are only approximate. However, the eigenvalues and their multiplicities are exact.

q_0 is in K_0, and every strongly connected component is reachable.[14] For $i = 0, 1, \ldots, k-1$, let ρ_i denote the Perron-Frobenius eigenvalue of the transition matrix $\Delta_{A[K_i,q_i]}$ of the automaton $A[K_i, q_i]$, where q_i is an arbitrary state in the set K_i, for $i = 1, \ldots, k-1$. Then, the automaton A is weakly state-$\mathcal{C}_=$-equiloaded, if and only if the following properties hold:

(i) The Perron-Frobenius eigenvalues corresponding to the strongly connected components of the graphical representation of the automaton A are all the same, i.e.,

$$\rho_0 = \rho_1 = \ldots = \rho_{k-1}.$$

(ii) The length of the longest path x_i in $\mathrm{Pth}[K_0, K_i, *]$, is equal for all i in $\{0, 1, \ldots, k-1\}$, or $\rho_i = 0$ for $i = 0, 1, \ldots, k-1$.

Proof. We shall prove each implication separately.

$\Rightarrow$: First, let us suppose that the property (i) does not hold. Then, indices i, j in $\{0, 1, \ldots, k-1\}$ exist, such that $\rho_i < \rho_j$. Moreover, let ρ_j be the greatest of all Perron-Frobenius eigenvalues. Let q_i in K_i be a state. Let ρ_{max} be a greatest of the Perron-Frobenius eigenvalues corresponding to the strongly connected components on the paths in $\mathrm{Pth}[K_0, K_i, *]$. It follows from Lemma 2.3.29 that

$$\#[q_i, \mathrm{Comp}(A, n)] = \Theta\left(n^r \cdot \rho_{max}^n\right),$$

for some r in $\mathbb{N}$.

Now, if $\rho_{max} < \rho_j$, then, by Lemma 2.3.29, the property

$$\#[q_j, \mathrm{Comp}(A, n)] = \Theta\left(n^{r'} \cdot \rho_j^n\right)$$

holds for some q_j in K_j and r' in $\mathbb{N}$, and the automaton A is obviously not weakly state-$\mathcal{C}_=$-equiloaded.

If $\rho_{max} = \rho_j$, then s in $\{0, 1, \ldots, k-1\}$ has to exist, such that $\rho_s = \rho_j = \rho_{max}$, and by Lemma 2.3.29, such that

$$\#[q_s, \mathrm{Comp}(A, n)] = \Omega\left(n^{r+1} \cdot \rho_{max}^n\right)$$

for some q_s in K_s. It is clear that the automaton A is not weakly state-$\mathcal{C}_=$-equiloaded.

Finally, let us suppose that the property (i) holds, and that the property (ii) does not hold. It clearly follows from Lemma 2.3.29 that the automaton A is not weakly state-$\mathcal{C}_=$-equiloaded.

$\Leftarrow$: This implication follows directly from Lemma 2.3.29.

The characterization is proved. □

Although we leave the characterizations of weakly state-$\mathcal{S}$-equiloaded DFA and DFAε open for $\mathcal{S}$ in $\{\mathcal{A}_=, \mathcal{C}_\le, \mathcal{A}_\le\}$, let us note that the above-presented characterization for $\mathcal{S} = \mathcal{C}_=$ and the methods used in its proof give us a sharp intuition also for the remaining cases. Thus, we may conjecture that a deterministic finite automaton is weakly state-$\mathcal{C}_\le$-equiloaded if and only if it is weakly state-$\mathcal{C}_=$-equiloaded. Furthermore, we suppose that the characterization of weak state-$\mathcal{A}_=$-equiloadedness will be the same as the characterization of weak state-$\mathcal{C}_=$-equiloadedness, with an additional requirement related to periodicity. Anyway, we leave the details as a subject for the later study.

[14] It is clearly possible to construct an equivalent DFAε satisfying the property of reachability to every DFAε by simply deleting the strongly connected components that are not reachable.

2.3.6 Relations between the Families of $\mathcal{S}$-Equiloaded Languages

In this subsection, we shall examine the mutual relations between the families of $\mathcal{S}$-equiloaded and weakly $\mathcal{S}$-equiloaded DFA-languages and DFAε-languages. Since the only families of such languages that have been studied up to now are the families $\mathscr{L}_{K-EQ-DFA}(\mathcal{A}_=)$, $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{A}_=)$, and $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{A}_=)$, most of the results presented in this subsection are completely new.

We shall start by presenting the result that shows a certain robustness of the definition of weak transition-$\mathcal{S}$-equiloadedness: we shall prove that the families of weakly transition-$\mathcal{S}$-equiloaded DFA-languages are the same, no matter what $\mathcal{S}$ from the set $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$ is chosen. The same property holds also for the families of weakly transition-$\mathcal{S}$-equiloaded DFAε-languages.

Theorem 2.3.32 The following identities hold:

1. $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{C}_=) = \mathscr{L}_{\delta-WEQ-DFA}(\mathcal{A}_=) = \mathscr{L}_{\delta-WEQ-DFA}(\mathcal{C}_\leq) = \mathscr{L}_{\delta-WEQ-DFA}(\mathcal{A}_\leq)$,
2. $\mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{C}_=) = \mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{A}_=) = \mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{C}_\leq) = \mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{A}_\leq)$.

Proof. Let L be a regular language, for which a deterministic finite automaton (with or without ε-transitions) A exists, such that $L(A) = L$ and the graphical representation of the automaton A does not contain any reachable directed cycle from which some accepting state is reachable. Then, clearly, a deterministic finite automaton A' accepting L exists, such that its graphical representation does not contain any reachable directed cycle – it clearly suffices to delete all states q of the automaton A, such that there is not any accepting state reachable from q in A.

The statement of the theorem is then a direct consequence of this fact, of Theorem 2.3.18, and of Theorem 2.3.19. □

Now we shall prove that, similarly as in the case of the strict $\mathcal{S}$-equiloadedness, the use of ε-transitions strengthens the computational power of weakly transition-$\mathcal{S}$-equiloaded deterministic finite automata, as well as transition-$\mathcal{S}$-equiloaded deterministic finite automata.

Theorem 2.3.33 The following strict inclusions hold for all $\mathcal{S}$ in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$:

1. $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{S}) \subsetneq \mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{S})$,
2. $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{S}) \subsetneq \mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{S})$.

Proof. According to Theorem 2.3.32, it is sufficient to prove the theorem for one arbitrary $\mathcal{S}$ in the set $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$. We shall prove the theorem for $\mathcal{S} = \mathcal{C}_=$.

Let us consider the language $L = \{a\}^+$. In the proof of Theorem 2.2.8, we have constructed a deterministic finite automaton with ε-transitions accepting L. It can be easily proved that this automaton is transition-$\mathcal{C}_=$-equiloaded. Thus, the language L is both in $\mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{C}_=)$ and in $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{C}_=)$.

However, on the other hand, the graphical representation of the minimal DFA accepting L (that can be easily constructed by the reader) is not strongly connected and, at the same time, contains a reachable directed cycle. Thus, by Theorem 2.3.20, the language L is not the member of the family $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{C}_=)$ and, as a consequence, nor of the family $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{C}_=)$. That is, the theorem is proved. □

In the following theorem we shall observe that, unlike in the case of weak transition-$\mathcal{S}$-equiloadedness, the families of transition-$\mathcal{S}$-equiloaded DFA(ε)-languages are not all the same for $\mathcal{S}$ in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$. However, in Theorem 2.3.35, we shall prove that the only difference between these families of languages is in finite languages – that is, these families of languages are *almost* the same.

Theorem 2.3.34 The following relations hold:

1. $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{C}_\leq) \subsetneq \mathscr{L}_{\delta-EQ-DFA}(\mathcal{C}_=) = \mathscr{L}_{\delta-EQ-DFA}(\mathcal{A}_=)$,

2. $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{C}_{\leq}) \subsetneq \mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{C}_{=}) = \mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{A}_{=})$,

3. $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{A}_{\leq}) \subsetneq \mathscr{L}_{\delta-EQ-DFA}(\mathcal{C}_{=}) = \mathscr{L}_{\delta-EQ-DFA}(\mathcal{A}_{=})$,

4. $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{A}_{\leq}) \subsetneq \mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{C}_{=}) = \mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{A}_{=})$.

5. The families $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{C}_{\leq})$ and $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{A}_{\leq})$ are incomparable.

6. The families $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{C}_{\leq})$ and $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{A}_{\leq})$ are incomparable.

Proof. First, we shall prove that $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{C}_{=}) = \mathscr{L}_{\delta-EQ-DFA}(\mathcal{A}_{=})$ and $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{C}_{=}) = \mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{A}_{=})$. Let L be a finite language. Then, clearly, a deterministic finite automaton with acyclic graphical representation exists, accepting L. However, deterministic finite automata with acyclic graphical representation are clearly both transition-$\mathcal{C}_{=}$-equiloaded and transition-$\mathcal{A}_{=}$-equiloaded. Thus, both identities hold for finite languages. Now, let L be an infinite language. Then, the graphical representation of every deterministic finite automaton accepting L has to contain at least one reachable directed cycle from which some accepting state is reachable. However, by Theorem 2.3.16, such an automaton is transition-$\mathcal{C}_{=}$-equiloaded if and only if it is transition-$\mathcal{A}_{=}$-equiloaded. Thus, the identities $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{C}_{=}) = \mathscr{L}_{\delta-EQ-DFA}(\mathcal{A}_{=})$ and $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{C}_{=}) = \mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{A}_{=})$ are proved.

Now, we shall prove the proper inclusions from claims 1 – 4. First, we shall at once prove the inclusions $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{C}_{\leq}) \subsetneq \mathscr{L}_{\delta-EQ-DFA}(\mathcal{C}_{=})$ and $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{C}_{\leq}) \subsetneq \mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{C}_{=})$. Let us consider the language $L_1 = \{ab\}$. The language L_1 is finite, and thus, clearly is both in $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{C}_{=})$, and in $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{C}_{=})$. We shall prove that there is not any transition-$\mathcal{C}_{\leq}$-equiloaded DFAε accepting L_1. This will also clearly imply that there is not any transition-$\mathcal{C}_{\leq}$-equiloaded DFA accepting L_1. For the purpose of contradiction, let us suppose that such a DFAε A exists. Let q be the first state of this automaton (in the direction of every computation path of the automaton A), such that there is a non-ε transition leading from q. Clearly, there has to be a transition $e = (q, a, q')$ on a leading from q to some other state q' – otherwise, there would not be any word in L_1 beginning with a, and that would be a contradiction. Moreover, there has to be a transition f on b reachable from q', such that for every computation path γ in $\mathrm{Comp}(A)$, the inequality $\#[f, \gamma] \leq 1$ holds, and $\#[f, \gamma] = 1$ implies also $\#[e, \gamma] = 1$ – otherwise, the word ab would either not be in L_1, or there would be some other word in L_1. This, togehter with our assumption of $\mathcal{C}_{\leq}$-equiloadedness, implies also that $\mathrm{Comp}(A)$ is finite. However, there clearly is a computation path γ' in $\mathrm{Comp}(A)$, such that $\#[e, \gamma] \geq 1$ and, at the same time, $\#[f, \gamma] = 0$. Thus, $\#[e, \mathrm{Comp}(A)] > \#[f, \mathrm{Comp}(A)]$. However, since $\mathrm{Comp}(A)$ is finite, there is a nonnegative integer n_0 in $\mathbb{N}$, such that $\mathrm{Comp}(A) = \mathrm{Comp}(A, \leq n)$ for all $n \geq n_0$. Thus, $\#[e, \mathrm{Comp}(A, \leq n)] > \#[f, \mathrm{Comp}(A, \leq n)]$ for all $n \geq n_0$, and that clearly implies that the inequality $B_A(\mathcal{C}_{\leq}) < 1$ holds. That contradicts our assumption that the automaton A is transition-$\mathcal{C}_{\leq}$-equiloaded.

Now, we shall simultaneously prove the inclusions $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{A}_{\leq}) \subsetneq \mathscr{L}_{\delta-EQ-DFA}(\mathcal{C}_{=})$, and $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{A}_{\leq}) \subsetneq \mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{C}_{=})$. Let us consider the language $L_2 = \{a, b, aa, ba, cc, dc\}$. Since the language L_2 is finite, it is clearly both in $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{C}_{=})$, and in $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{C}_{=})$. Again, we shall prove that there is not any transition-$\mathcal{A}_{\leq}$-equiloaded DFAε accepting L_2, and this will imply the same consequence also for DFA. For the purpose of contradiction, let us suppose that a transition-$\mathcal{A}_{\leq}$-equiloaded DFAε A exists, such that $L(A) = L_2$. Let q be the first (in the direction of every computation path of the automaton) state of the automaton A, such that there is at least one non-ε transition leading from A. Then, there has to be a transition e_x on each character x in the set $\{a, b, c, d\}$ leading from q – otherwise, the property $L(A) = L_2$ would not be satisfied. Moreover, it is clear that the property $(q, w) \vdash^{+} (q, \varepsilon)$ cannot hold for any w in Σ^{+} – otherwise the language $L(A)$ would be either empty or infinite, and that would contradict $L(A) = L_2$. Thus, each of these transitions is used exactly once in accepting computations on words beginning with the corresponding character, and is not used in other accepting computations. Thus, for each x in $\{a, b, c, d\}$,

$$\#[x, \mathrm{Acc}(A)] = |\{w \in L_2 \mid w[1] = x\}|.$$

However, there are two words in L_2 beginning with a and b, but only one word in L beginning with c and d. Thus,

$$\#[a, \mathrm{Acc}(A)] > \#[c, \mathrm{Acc}(A)],$$

and since the accepted language L_2 is finite, for n greater than some n_0 in $\mathbb{N}$, we have also

$$\#[a, \mathrm{Acc}(A, \leq n)] > \#[c, \mathrm{Acc}(A, \leq n)].$$

That is, $B_A(\mathcal{A}_\leq) < 1$, and that contradicts our assumption that the automaton A is transition-$\mathcal{A}_\leq$-equiloaded.

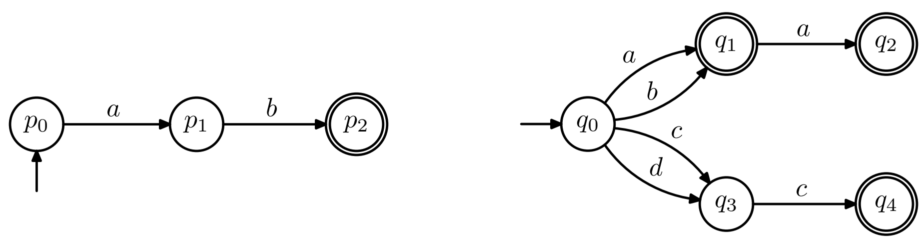
It remains to prove the claims from points 5 and 6. These claims will be obviously proved, if we construct a transition-$\mathcal{A}_\leq$-equiloaded DFA (that can be obviously viewed also as a DFAε) $A_1 = (K_1, \Sigma_1, \delta_1, p_0, F_1)$, such that $L(A_1) = L_1$, and a transition-$\mathcal{C}_\leq$-equiloaded DFA $A_2 = (K_2, \Sigma_2, \delta_2, q_0, F_2)$, such that $L(A_2) = L_2$. We shall define these automata as follows: $K_1 = \{p_0, p_1, p_2\}$, $\Sigma_1 = \{a, b\}$, $F = \{p_2\}$, and

$$\delta_1(p_0, a) = p_1,$$
$$\delta_1(p_1, b) = p_2.$$

As it can be easily observed,

$$\#[e, \mathrm{Acc}(A, \leq n)] = 1$$

for every transition e in D_{A_1} and all n in $\mathbb{N}$, such that $n \geq 2$. Thus, the automaton A_1 is transition-$\mathcal{A}_\leq$-equiloaded.



(a) The automaton A_1. (b) The automaton A_2.

Figure 2.5: The transition-$\mathcal{A}_\leq$-equiloaded automaton A_1 accepting the language L_1 and the transition-$\mathcal{C}_\leq$-equiloaded automaton A_2 accepting the language L_2.

For the automaton A_2, we shall define $K_2 = \{q_0, q_1, q_2, q_3, q_4\}$, $\Sigma = \{a, b, c, d\}$, $F = \{q_1, q_2, q_4\}$, and

$$\delta_2(q_0, a) = q_1, \quad \delta_2(q_0, b) = q_1, \quad \delta_2(q_0, c) = q_3,$$
$$\delta_2(q_0, d) = q_3, \quad \delta_2(q_1, a) = q_2, \quad \delta_2(q_3, c) = q_4.$$

The reader may easily convince himself that $L(A_2) = L_2$. Moreover, as can be clearly observed,

$$\#[e, \mathrm{Comp}(A, \leq n)] = 2$$

for every transition e in D_{A_2} and all n in $\mathbb{N}$, such that $n \geq 2$. Thus, the automaton A_2 is transition-$\mathcal{C}_\leq$-equiloaded. That is, the theorem is proved. $\square$

Theorem 2.3.35 For a given family of languages $\mathscr{L}$, let $\mathscr{L}^{inf}$ denote the family of all infinite languages in $\mathscr{L}$. Then, the following identities hold:

1. $\mathscr{L}^{inf}_{\delta-EQ-DFA}(\mathcal{C}_=) = \mathscr{L}^{inf}_{\delta-EQ-DFA}(\mathcal{A}_=) = \mathscr{L}^{inf}_{\delta-EQ-DFA}(\mathcal{C}_\leq) = \mathscr{L}^{inf}_{\delta-EQ-DFA}(\mathcal{A}_\leq)$,

2. $\mathscr{L}^{inf}_{\delta-EQ-DFA\varepsilon}(\mathcal{C}_=) = \mathscr{L}^{inf}_{\delta-EQ-DFA\varepsilon}(\mathcal{A}_=) = \mathscr{L}^{inf}_{\delta-EQ-DFA\varepsilon}(\mathcal{C}_\leq) = \mathscr{L}^{inf}_{\delta-EQ-DFA\varepsilon}(\mathcal{A}_\leq)$.

Proof. Let L in $\mathscr{R}^{inf}$ be an infinite regular language. Then, since L is infinite, the graphical representation of every deterministic finite automaton accepting L has to contain at least one reachable directed cycle from which some accepting state is reachable. Thus, both claims are direct consequences of Theorem 2.3.16. □

In Theorem 2.3.38, we shall prove that the families of transition-$\mathcal{S}$-equiloaded DFA-languages are the *proper* subsets of the corresponding families of weakly transition-$\mathcal{S}$-equiloaded DFA-languages, for $\mathcal{S}$ in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$. In [25], we have already proved this for $\mathcal{S} = \mathcal{A}_=$. We shall observe that this proof can be easily generalized also to other choices of $\mathcal{S}$.

For DFAε, the situation seems to be more complicated, and we shall leave the problem of relation between the families $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{S})$ and $\mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{S})$ as an open problem.

We shall present the main idea of the proof for DFA in Lemma 2.3.37. The key is to find an efficient proof method for proving that a given language is *not* in the family of transition-$\mathcal{S}$-equiloaded DFA-languages. Unlike the proofs presented up to now, this proof method is required to apply also to languages that are weakly transition-$\mathcal{S}$-equiloaded DFA-languages. The method based on Lemma 2.3.37 will prove to be sufficient for the purpose of proving the above mentioned relation.

However, before we state this lemma, we shall present a notation that we shall use in Lemma 2.3.37, and subsequently also in the proof of Theorem 2.3.38.

Notation 2.3.36 Let $A = (K, \Sigma, \delta, q_0, F)$ be a DFA, let c in Σ be a character. Then, by D^c_A, we shall denote the set of all transitions of the automaton A on the character c, i.e.,

$$D^c_A = \{(q, c, q') \in K \times \{c\} \times K \mid (q, c, q') \in D_A\}.$$

If A is clear from the context, we shall write only D^c instead of D^c_A.

Lemma 2.3.37 Let L in $\mathscr{R}$ be a regular language over an alphabet Σ. Let A_1, A_2 be DFA (without ε-transitions), such that $L(A_1) = L(A_2) = L$. Then, the identity

$$\sum_{e \in D^c_{A_1}} \#[e, \mathrm{Acc}(A_1, n)] = \sum_{f \in D^c_{A_2}} \#[f, \mathrm{Acc}(A_2, n)] = \sum_{w \in L \cap \Sigma^n} \#_c(w).$$

holds for every character c in Σ and all n in $\mathbb{N}$.

Proof. The lemma is a direct consequence of the fact that if a word w is in $L \cap \Sigma^n$, then the accepting computation path γ on the word w is both in $\mathrm{Acc}(A_1, n)$, and in $\mathrm{Acc}(A_2, n)$. □

Theorem 2.3.38 For every $\mathcal{S}$ in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$, the strict inclusion

$$\mathscr{L}_{\delta-EQ-DFA}(\mathcal{S}) \subsetneq \mathscr{L}_{\delta-WEQ-DFA}(\mathcal{S})$$

holds.

Proof. The proof will follow the idea of the proof for $\mathcal{S} = \mathcal{A}_=$, presented in [25], and will make use of Lemma 2.3.37.

The inclusion $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{S}) \subseteq \mathscr{L}_{\delta-WEQ-DFA}(\mathcal{S})$ is trivially satisfied, since every transition-$\mathcal{S}$-equiloaded automaton is a weakly transition-$\mathcal{S}$-equiloaded automaton as well.

To prove that the inclusion is proper, let us consider a language $L = \{a, bb\}^*$. There is a deterministic finite automaton without ε-transitions, $A = (K, \Sigma, \delta, q_0, F)$, accepting L, and defined as follows: $K = \{q_0, q_1\}$, $\Sigma = \{a, b\}$, $F = \{q_0\}$, and

$$\delta(q_0, a) = q_0,$$
$$\delta(q_0, b) = q_1,$$
$$\delta(q_1, b) = q_0.$$

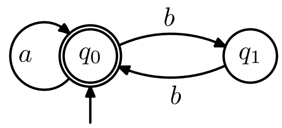


Figure 2.6: The automaton A with strongly connected graphical representation, accepting the language $L = \{a, bb\}^*$.

The graphical representation of the automaton A is depicted in Figure 2.6. The claim $L(A) = L$ is considered to be obvious. Since the graphical representation of the automaton A is strongly connected, the DFA A is weakly transition-$\mathcal{S}$-equiloaded for all $\mathcal{S}$ in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\le, \mathcal{A}_\le\}$. That is, L is in $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{S})$ for all $\mathcal{S}$ in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\le, \mathcal{A}_\le\}$.

However, on the other hand, by applying the method presented in Section 2.1, it can be easily computed that

$$\begin{aligned}
\#[(q_0, a, q_0), \mathrm{Acc}(A, n)] &= \frac{1+\sqrt{5}}{10} \cdot n \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n + \frac{2}{5\sqrt{5}} \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n + \\
&\quad + \frac{1-\sqrt{5}}{10} \cdot n \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n - \frac{2}{5\sqrt{5}} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n, \\
\#[(q_0, b, q_1), \mathrm{Acc}(A, n)] &= \frac{1}{5} \cdot n \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{5\sqrt{5}} \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n + \\
&\quad + \frac{1}{5} \cdot n \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n + \frac{1}{5\sqrt{5}} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n, \\
\#[(q_1, b, q_0), \mathrm{Acc}(A, n)] &= \frac{1}{5} \cdot n \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{5\sqrt{5}} \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n + \\
&\quad + \frac{1}{5} \cdot n \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n + \frac{1}{5\sqrt{5}} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n.
\end{aligned}$$

Now, for the purpose of contradiction, let us suppose that a transition-$\mathcal{A}_=$-equiloaded DFA A' exists, such that $L(A') = L$. Then, by Lemma 2.3.37 and by the assumption that $B_{A'} = 1$, we have

$$\#[e, \mathrm{Acc}(A', n)] = \frac{1+\sqrt{5}}{10 \cdot |D^a_{A'}|} \cdot n \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n + o\left(n \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n\right)$$

for all e in $D^a_{A'}$, and

$$\#[f, \mathrm{Acc}(A', n)] = \frac{2}{5 \cdot |D^b_{A'}|} \cdot n \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n + o\left(n \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n\right)$$

for all f in $D^b_{A'}$. Thus, obviously, from the assumption that the automaton A is transition-$\mathcal{A}_=$-equiloaded, we have

$$\frac{1+\sqrt{5}}{10 \cdot |D^a_{A'}|} = \frac{2}{5 \cdot |D^b_{A'}|},$$

i.e.,

$$\frac{1+\sqrt{5}}{10} \cdot |D^b_{A'}| = \frac{2}{5} \cdot |D^a_{A'}|.$$

But this equation cannot hold, since $|D^a_{A'}|$ and $|D^b_{A'}|$ are both nonnegative integers, since $\frac{2}{5}$ is a rational number, and since $\frac{1+\sqrt{5}}{10}$ is an irrational number. Thus, there is not any transition-$\mathcal{A}_=$-equiloaded DFA accepting L, i.e., L is not in $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{A}_=)$. However, since the language L is infinite, it follows from Theorem 2.3.35, that L is not in $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{S})$ for any $\mathcal{S}$. That is, the theorem is proved. □

We leave an analogous property for the families of $\mathcal{S}$-equiloaded resp. weakly $\mathcal{S}$-equiloaded DFAε-languages as an open problem.

Open Problem 2.3.39 The inclusion $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{S}) \subseteq \mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{S})$ holds trivially for all $\mathcal{S}$ in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$. Is this inclusion proper or does $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{S}) = \mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{S})$ hold for $\mathcal{S}$ in the set $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$?

Example 2.3.40 A transition-$\mathcal{S}$-equiloaded DFAε can be found for a surprisingly high number of weakly transition-$\mathcal{S}$-equiloaded languages (however, to little examples have been worked out yet, to conjecture the equality of the families).

First, we shall observe that the counterexample used for the case of DFA cannot be used in the case of DFAε, i.e., that there is a transition-$\mathcal{A}_=$-equiloaded DFAε accepting the language $L = \{a, bb\}^*$. We shall define such automaton as follows: $A = (K, \Sigma, \delta, q_0, F)$, $K = \{q_0, q_1, q_2\}$, $F = \{q_0\}$, and

$$\delta(q_0, a) = q_2, \qquad \delta(q_0, b) = q_1, \qquad \delta(q_1, b) = q_0, \qquad \delta(q_2, \varepsilon) = q_0.$$

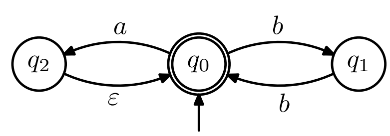


Figure 2.7: The transition-$\mathcal{A}_=$-equiloaded DFAε A accepting the language $L = \{a, bb\}^*$.

The reader may convince himself that the automaton A is transition-$\mathcal{A}_=$-equiloaded, by computing the transition-equiloadedness $\mathcal{A}_=$-measure, using the method presented in Section 2.1. That is, L is in $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{A}_=)$, and since the language L is infinite, it follows from Theorem 2.3.35 that L is in $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{S})$, for all $\mathcal{S}$ in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$.

Example 2.3.41 Further, let us consider a language L accepted by a deterministic finite automaton $A = (K, \Sigma, \delta, q_0, F)$ defined as follows: $K = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{a, b\}$, $F = \{q_0\}$, and

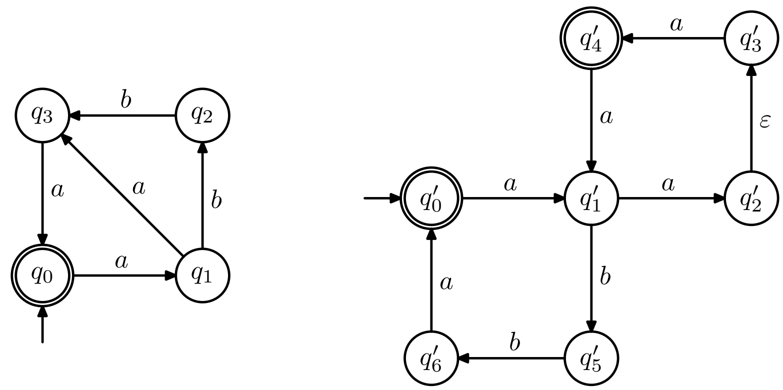
$$\delta(q_0, a) = q_1, \qquad \delta(q_1, a) = q_3, \qquad \delta(q_1, b) = q_2, \qquad \delta(q_2, b) = q_3, \qquad \delta(q_3, a) = q_0.$$

Clearly, the language accepted by the automaton A is the same as the language accepted by a DFAε defined as follows: $A' = (K', \Sigma', \delta', q'_0, F')$, $K' = \{q'_0, q'_1, \ldots, q'_6\}$, $\Sigma' = \{a, b\}$, $F' = \{q'_0, q'_4\}$, and

$$\begin{aligned} &\delta'(q'_0, a) = q'_1, \qquad \delta'(q'_1, a) = q'_2, \qquad \delta'(q'_1, b) = q'_5, \qquad \delta'(q'_2, \varepsilon) = q'_3, \\ &\delta'(q'_3, a) = q'_4, \qquad \delta'(q'_4, a) = q'_1, \qquad \delta'(q'_5, b) = q'_6, \qquad \delta'(q'_6, a) = q'_0. \end{aligned}$$

By computing the transition-equiloadedness $\mathcal{A}_=$-measure, it can be verified that the automaton A' is transition-$\mathcal{A}_=$-equiloaded. Thus, L is in $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{A}_=)$, and since the language L is infinite, it follows from Theorem 2.3.35 that L is in $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{S})$, for all $\mathcal{S}$ in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$.

Now, let us consider the families of state-$\mathcal{S}$-equiloaded and weakly state-$\mathcal{S}$-equiloaded languages. Let us note that a theorem analogous to Theorem 2.3.33 holds also for the families of



(a) The automaton A. (b) The automaton A'.

Figure 2.8: The automaton A, and the transition-$\mathcal{A}_=$-equiloaded automaton A', such that $L(A') = L(A)$.

(weakly) state-$\mathcal{S}$-equiloaded languages. Since exactly the same counterexample $L = \{a\}^+$ may be used to prove the theorem,[15] we shall omit the proof.

Theorem 2.3.42 The following strict inclusions hold for all $\mathcal{S}$ in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$:

1. $\mathscr{L}_{K-WEQ-DFA}(\mathcal{S}) \subsetneq \mathscr{L}_{K-WEQ-DFA\varepsilon}(\mathcal{S})$,
2. $\mathscr{L}_{K-EQ-DFA}(\mathcal{S}) \subsetneq \mathscr{L}_{K-EQ-DFA\varepsilon}(\mathcal{S})$.

We shall leave open the relations between the families of (weakly) state-$\mathcal{S}$-equiloaded languages for different choices of $\mathcal{S}$.

Finally in this subsection, let us consider the relations between $\mathcal{S}$-equiloaded and strictly $\mathcal{S}$-equiloaded DFA(ε)-languages, for $\mathcal{S} = \mathcal{C}_=$ and $\mathcal{S} = \mathcal{A}_=$.

Theorem 2.3.43 The following strict inclusions hold:

1. $\mathscr{L}_{K-SEQ-DFA}(\mathcal{C}) \subsetneq \mathscr{L}_{K-EQ-DFA}(\mathcal{C}_=)$,
2. $\mathscr{L}_{\delta-SEQ-DFA}(\mathcal{C}) \subsetneq \mathscr{L}_{\delta-EQ-DFA}(\mathcal{C}_=)$,
3. $\mathscr{L}_{K-SEQ-DFA}(\mathcal{A}) \subsetneq \mathscr{L}_{K-EQ-DFA}(\mathcal{A}_=)$,
4. $\mathscr{L}_{\delta-SEQ-DFA}(\mathcal{A}) \subsetneq \mathscr{L}_{\delta-EQ-DFA}(\mathcal{A}_=)$,
5. $\mathscr{L}_{K-SEQ-DFA\varepsilon}(\mathcal{C}) \subsetneq \mathscr{L}_{K-EQ-DFA\varepsilon}(\mathcal{C}_=)$,
6. $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}(\mathcal{C}) \subsetneq \mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{C}_=)$,
7. $\mathscr{L}_{K-SEQ-DFA\varepsilon}(\mathcal{A}) \subsetneq \mathscr{L}_{K-EQ-DFA\varepsilon}(\mathcal{A}_=)$,
8. $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}(\mathcal{A}) \subsetneq \mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{A}_=)$.

Proof. The inclusions are a direct consequence of Theorem 1.6.2. Let us prove that these inclusions are proper. Let us consider the languages $L_1 = \{a\}^* \cdot \{b\}^*$ and $L_2 = \{a, b\}^*$. It is an easy exercise to prove that L_1 is not in $\mathscr{L}_{K-SEQ-DFA}(\mathcal{C})$, nor in $\mathscr{L}_{K-SEQ-DFA}(\mathcal{A})$ and that it is in $\mathscr{L}_{K-EQ-DFA}(\mathcal{C}_=)$, as well as in $\mathscr{L}_{K-EQ-DFA}(\mathcal{A}_=)$. Similarly, it is easy to prove that L_2 is not in $\mathscr{L}_{\delta-SEQ-DFA}(\mathcal{C})$, nor in $\mathscr{L}_{\delta-SEQ-DFA}(\mathcal{A})$, and that it is both in $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{C}_=)$, and in $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{A}_=)$. □

2.3.7 Closure Properties

In this subsection, we shall examine some of the closure properties of the families of $\mathcal{S}$-equiloaded and weakly $\mathcal{S}$-equiloaded languages. We shall omit proofs of the results that have been already known (these proofs may be found in the theses [26], [27] and [25]) and prove only the new closure properties.

[15] It is easy to prove that this language is not in $\mathscr{L}_{K-WEQ-DFA}(\mathcal{S})$ by using the Myhill-Nerode theorem [32].

Closure Properties of the Families of (Weakly) State-$\mathcal{S}$-Equiloaded DFA(ε)-Languages

For the case of the families of state-$\mathcal{S}$-equiloaded and weakly state-$\mathcal{S}$-equiloaded DFA(ε)-languages, we leave most of their closure properties open. In the following theorem, we shall only restate the closure properties of the family $\mathscr{L}_{K-EQ-DFA}(\mathcal{A}_{=})$, proved already in [26] and [27]. The main reason for this is our belief that these closure properties will be less painfully provable after slightly extending the theory from the previous subsection (it will be possible to prove more closure properties at once, and thus the need to examine each case separately will be eliminated).

Theorem 2.3.44 The family $\mathscr{L}_{K-EQ-DFA}(\mathcal{A}_{=})$ is not closed under concatenation, union, intersection, complementation, reversal, homomorphism, and inverse homomorphism.

Closure Properties of the Families of (Weakly) Transition-$\mathcal{S}$-Equiloaded DFA(ε)-Languages

In what follows, we shall prove some of the closure properties of the families of transition-$\mathcal{S}$-equiloaded and weakly transition-$\mathcal{S}$-equiloaded DFA(ε)-languages. The results obtained in the previous subsection enable us to prove the closure properties for relatively large numbers of families of languages at once. Thus, our knowledge of the closure properties is better than in the case of state-$\mathcal{S}$-equiloaded and weakly state-$\mathcal{S}$-equiloaded languages.

Theorem 2.3.45 Let $\mathcal{S}$ be in $\{\mathcal{C}_{=}, \mathcal{A}_{=}, \mathcal{C}_{\leq}, \mathcal{A}_{\leq}\}$. The families $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{S})$, $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{S})$, $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{S})$, and $\mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{S})$ are not closed under concatenation.

Proof. The languages $L_1 = \{a\}^*$ and $L_2 = \{b\}$ are clearly in all of these families. However, the language $L_1 \cdot L_2 = \{a\}^* \cdot \{b\}$ is in none of these families. This can be easily proved by observing that in every DFAε accepting $L_1 \cdot L_2$, a b-transition e has to exist, such that it is used at most once in every computation path. However, a computation path of arbitrary length has to exist in every such automaton. □

Theorem 2.3.46 Let $\mathcal{S}$ be in $\{\mathcal{C}_{=}, \mathcal{A}_{=}, \mathcal{C}_{\leq}, \mathcal{A}_{\leq}\}$. The families $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{S})$, $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{S})$, $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{S})$, and $\mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{S})$ are not closed under union.

Proof. The same counterexample can be used as in the proof of Theorem 2.3.45. □

Theorem 2.3.47 Let $\mathcal{S}$ be in $\{\mathcal{C}_{=}, \mathcal{A}_{=}, \mathcal{C}_{\leq}, \mathcal{A}_{\leq}\}$. The families $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{S})$, $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{S})$, $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{S})$, and $\mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{S})$ are not closed under intersection.

Proof. Let us consider the languages $L_1 = \{aa, bc\}^* \cdot \{b\}$ and $L_2 = \{aa, bd\}^* \cdot \{b\}$. It can be easily observed that these languages are in all of the families from the statement of the theorem. However, their intersection, the language $L_1 \cap L_2 = \{aa\}^* \cdot \{b\}$, is clearly in none of these families. This can be proved by similar reasoning as in the proof of Theorem 2.3.45. □

Theorem 2.3.48 Let $\mathcal{S}$ be in $\{\mathcal{C}_{=}, \mathcal{A}_{=}, \mathcal{C}_{\leq}, \mathcal{A}_{\leq}\}$. The families $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{S})$, $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{S})$, $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{S})$, and $\mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{S})$ are not closed under complementation.

Proof. Let us consider the language $L = \{a\}$. This language clearly is in all of the families from the statement of the theorem. We shall prove that the language $L^C = \{\varepsilon\} \cup \{a^n \mid n \geq 2\}$ is not in any of these families.

For the purpose of contradiction, let us suppose that L^C is in some of this families. Since the language L^C is infinite, a DFAε $A = (K, \Sigma, \delta, q_0, F)$ with a strongly connected graphical representation has to exist, such that $L(A) = L^C$. Clearly, the relation $(q_0, a) \vdash^* (q, \varepsilon)$ cannot hold for any accepting state q in F. However, since the automaton A is strongly connected, a positive integer k exists, such that $(q_0, a^k) \vdash^* (q_0, \varepsilon)$. This implies that also the relation $(q_0, a^{k+1}) \vdash^* (q, \varepsilon)$ cannot hold for any accepting state q, and that implies that a^{k+1} is not in L^C. However, this is a contradiction since k is positive. □

Theorem 2.3.49 Let $\mathcal{S}$ be in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$. The families $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{S})$ and $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{S})$ are not closed under closure.

Proof. Corollary of a theorem proved in [25] (the counterexample used in the proof for $\mathcal{S} = \mathcal{A}_=$ can be easily shown to apply also for the remaining choices of $\mathcal{S}$). □

Open Problem 2.3.50 Let $\mathcal{S}$ be a parameter in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$. Are the families $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{S})$ and $\mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{S})$ closed under closure?

Theorem 2.3.51 Let $\mathcal{S}$ be in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$. The families $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{S})$ and $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{S})$ are not closed under positive closure.

Proof. Corollary of a theorem proved in [25]. □

Open Problem 2.3.52 Let $\mathcal{S}$ be a parameter in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$. Are the families $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{S})$ and $\mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{S})$ closed under positive closure?

Theorem 2.3.53 Let $\mathcal{S}$ be in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$. The families $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{S})$ and $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{S})$ are not closed under reversal.

Proof. Corollary of a theorem proved in [25]. □

Open Problem 2.3.54 Let $\mathcal{S}$ be a parameter in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$. Are the families $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{S})$ and $\mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{S})$ closed under reversal?

Theorem 2.3.55 Let $\mathcal{S}$ be in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$. The families $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{S})$ and $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{S})$ are not closed under homomorphism.

Proof. Corollary of a theorem proved in [25]. □

Open Problem 2.3.56 Let $\mathcal{S}$ be a parameter in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$. Are the families $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{S})$ and $\mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{S})$ closed under homomorphism?

Theorem 2.3.57 Let $\mathcal{S}$ be in $\{\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq\}$. The families $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{S})$, $\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{S})$, $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{S})$, and $\mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{S})$ are not closed under inverse homomorphism.

Proof. Let us consider the language $L = \{b\}$ (clearly, L is in all of the families from the statement of the theorem) and the homomorphism h defined by

$$h(a) = \varepsilon,$$
$$h(b) = b.$$

Clearly,

$$h^{-1}(L) = \{w \in \{a, b\}^* \mid \#_b(w) = 1\}.$$

We shall prove that $h^{-1}(L)$ is not in any of the families from the statement of the theorem. For the purpose of contradiction, let us suppose that it is in at least one of these families. Since the language $h^{-1}(L)$ is infinite, this implies that a DFAε A with strongly connected graphical representation has to exist, such that $L(A) = h^{-1}(L)$. However, this automaton has to have at least one b-transition and at least one accepting state, and from that a contradiction is easy to reach. □

Chapter 3

Strictly $\mathcal{S}$-Equiloaded Deterministic One-Counter Automata

In this chapter, we shall study the families of strictly $\mathcal{S}$-equiloaded deterministic one-counter automata and the corresponding families of languages, for $\mathcal{S} = \mathcal{C}$, $\mathcal{S} = \mathcal{A}$, and $\mathcal{S} = \mathcal{E}$. Equiloaded one-counter automata have not been studied yet. That is, all results presented in this chapter are new.

3.1 Examples of Strictly $\mathcal{S}$-Equiloaded DOCA-Languages

In order to build an adequate picture of the computational power of strictly $\mathcal{S}$-equiloaded one-counter automata, we shall work out several examples.

Example 3.1.1 In this example, we shall present three examples of languages in the families $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$, and $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$. Let us consider the following three regular languages: $L_1 = \{a\}^*\{b\}^*$, $L_2 = \{a\}^*\{\varepsilon, b\}$, and $L_3 = \{a\}^* \cup \{b\}$. For each of these languages, we shall construct a deterministic one-counter automaton that is both strictly state-$\mathcal{C}$-equiloaded, and strictly state-$\mathcal{A}$-equiloaded.

We shall define a deterministic one-counter automaton $A_1 = (K_1, \Sigma_1, \delta_1, q_{0,1}, F_1)$ accepting the language L_1 as follows: $K_1 = \{q_0\}$, $\Sigma_1 = \{a, b\}$, $q_{0,1} = q_0$, $F_1 = \{q_0\}$, and

$$\delta_1(q_0, a, 0) = (q_0, 0), \qquad \delta_1(q_0, b, 0) = (q_0, +1), \qquad \delta_1(q_0, b, 1) = (q_0, 0).$$

The claim $L(A_1) = L_1$ is considered to be obvious. The strict state-$\mathcal{S}$-equiloadedness of the automaton A_1 for $\mathcal{S}$ in $\{\mathcal{C}, \mathcal{A}\}$ is a direct consequence of the fact that the automaton has only one state.

Now, let us construct a deterministic one-counter automaton $A_2 = (K_2, \Sigma_2, \delta_2, q_{0,2}, F_2)$, such that $L(A_2) = L_2$. We shall define the automaton as follows: $K_2 = \{q_0\}$, $\Sigma_2 = \{a, b\}$, $q_{0,2} = q_0$, $F_2 = \{q_0\}$, and

$$\delta_2(q_0, a, 0) = (q_0, 0), \qquad \delta_2(q_0, b, 0) = (q_1, +1).$$

Once again, the claim $L(A_2) = L_2$ is considered to be obvious, and the strict state-$\mathcal{S}$-equiloadedness of the automaton A_2 for $\mathcal{S}$ in $\{\mathcal{C}, \mathcal{A}\}$ follows from the fact that there is only one state in K_2.

Finally, we shall construct a deterministic one-counter automaton $A_3 = (K_3, \Sigma_3, \delta_3, q_{0,3}, F_3)$ accepting the language L_3. We shall define the automaton as follows: $K_3 = \{q_0, q_1\}$, $\Sigma_3 = \{a, b\}$, $q_{0,3} = q_0$, $F_3 = \{q_0, q_1\}$, and

$$\begin{aligned} \delta_3(q_0, a, 0) &= (q_1, +1), & \delta_3(q_0, b, 0) &= (q_1, 0), \\ \delta_3(q_0, a, 1) &= (q_1, 0), & \delta_3(q_1, a, 1) &= (q_0, 0). \end{aligned}$$

The claim $L(A_3) = L_3$ is considered to be obvious. The strict state-$\mathcal{S}$-equiloadedness of the automaton A_3 for $\mathcal{S}$ in $\{\mathcal{C}, \mathcal{A}\}$ follows from the fact that every computation path γ of the automaton A_3 alternates between states q_0 and q_1. Thus, as a consequence we have

$$|\#[q_0, \gamma] - \#[q_1, \gamma]| \leq 1$$

for every computation path γ, and the automaton A_3 is strictly state-$\mathcal{C}$-equiloaded, as well as strictly state-$\mathcal{A}$-equiloaded.

In the construction of each of the three automata from the previous example, we have exploited the fact that we can use one state of a deterministic one-counter automaton to act like two states of a deterministic finite automaton – if we use only two values of the counter, 0 and 1, then we may view the deterministic one-counter automaton with the set of states K as a deterministic finite automaton with the set of states $K \times \{0, 1\}$. However, the differences between the number of uses of a state q with the counter value 0 and the number of uses of the state q with the counter value 1 have no effect on the strict equiloadedness: the only thing we care about is the overall number of uses of the state q (without taking the counter value into consideration). Thus, the families of strictly state-$\mathcal{S}$-equiloaded DOCA-languages contain also some regular languages that are not strictly state-$\mathcal{S}$-equiloaded DFAε-languages (however, as we shall observe later, strictly state-$\mathcal{S}$-equiloaded DOCA may accept also nonregular languages).

This method cannot be used for strict transition-$\mathcal{S}$-equiloadedness. Unlike in the case of states, the transition from a state q on some c in $\Sigma \cup \{\varepsilon\}$ is a different transition for a counter value 0, and for a counter value greater than 0. However, there are other methods that can be used in the construction of strictly transition-$\mathcal{S}$-equiloaded DOCA for languages that are not in $\mathscr{L}_{\delta-SEQ-DFA}$ and $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. We shall present one such method in the following example.

Example 3.1.2 Let us consider the language $L = \{ab, ba\}^*$. We shall observe that this language is both in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C})$, and in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$. The main idea of the construction of a strictly transition-$\mathcal{S}$-equiloaded DOCA $A = (K, \Sigma, \delta, q_0, F)$ accepting L is as follows: each computation of the automaton A is required to begin with a sequence of ε-transitions that increases the counter from 0 to 5. Afterwards, the computation is required to "decide" (in the state q_5) between two cycles: after passing the first of the cycles, the character a is read, and the counter is decreased by the value of 2 (if the counter value is less than 2 at the beginning of the cycle, the computation gets stuck during the cycle). After passing the second cycle, the character b is read, and the counter is decreased by the value of 3. The computation is allowed to proceed from the state q_5 to the initial state only if the counter value 0 is reached. This is clearly the case if and only if each of the cycles is passed exactly once.

Formally, we shall construct the deterministic one-counter automaton A as follows: we shall define $K = \{q_0, q_1, \ldots, q_8\}$, $\Sigma = \{a, b\}$, $F = \{q_0\}$, and

$$\begin{array}{llll}
\delta(q_0, \varepsilon, 0) = (q_1, +1), & \delta(q_1, \varepsilon, 1) = (q_2, +1), & \delta(q_2, \varepsilon, 1) = (q_3, +1), & \delta(q_3, \varepsilon, 1) = (q_4, +1), \\
\delta(q_4, \varepsilon, 1) = (q_5, +1), & \delta(q_5, \varepsilon, 0) = (q_0, 0), & \delta(q_5, a, 1) = (q_6, -1), & \delta(q_5, b, 1) = (q_7, -1), \\
\delta(q_6, \varepsilon, 1) = (q_5, -1), & \delta(q_7, \varepsilon, 1) = (q_8, -1), & \delta(q_8, \varepsilon, 1) = (q_5, -1). &
\end{array}$$

The reader may easily convince himself that $L(A) = L$. The automaton is strictly transition-$\mathcal{S}$-equiloaded for $\mathcal{S}$ in $\{\mathcal{C}, \mathcal{A}\}$, since, for the above explained reasons, the property

$$|\#[e, \gamma] - \#[f, \gamma]| \leq 3$$

has to hold for all e, f in D, and for every computation path γ (for accepting computation paths, the constant 3 may be replaced by 0).

Example 3.1.3 Let us consider the deterministic one-counter automaton A from the previous example, once again. Clearly,

$$N(A) = \{ab, ba\}^*\{\varepsilon, aaa, aab, bb\} =: L'.$$

We have already noted that the automaton A is strictly transition-$\mathcal{C}$-equiloaded, and thus also strictly transition-$\mathcal{E}$-equiloaded. Thus, the language L' is both in $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$, and in $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E})$.

In the final three examples of this section, we shall return to strictly state-$\mathcal{S}$-equiloaded DOCA-languages. We shall show that among these languages, there are also some nonregular languages. Moreover, we shall observe that in the families of languages $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$ and $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$, there are also some languages that are not prefix-dense.

Example 3.1.4 In this example, we shall show that the (obviously nonregular) language

$$L = \{a^n b^n \mid n \geq 0\}$$

is in $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$. Let us define a deterministic one-counter automaton $A = (K, \Sigma, \delta, q_0, F)$ as follows: $K = \{q_0, q_1\}$, $\Sigma = \{a, b\}$, $F = \emptyset$ (since we are interested in the language accepted by empty memory, the set F is irrelevant), and

$$\begin{aligned} \delta(q_0, a, 0) &= (q_0, +1), \qquad & \delta(q_0, a, 1) &= (q_0, +1), \\ \delta(q_0, b, 1) &= (q_1, -1), \qquad & \delta(q_1, b, 1) &= (q_1, -1). \end{aligned}$$

We consider the claim $L(A) = L$ to be obvious. Moreover, in every computation path γ of the automaton A, accepting by empty memory, the following property clearly holds:

$$\#[q_1, \gamma] = \#[q_0, \gamma] - 1.$$

Thus, the automaton A is strictly state-$\mathcal{E}$-equiloaded. As a consequence, the language L is in $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$.

As shown in Example 1.7.6, the language L is not prefix-dense. Thus, it is neither in the family $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$, nor in the family $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$.

Example 3.1.5 In this example, we shall observe that also the families $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$ and $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$ contain some nonregular languages. Let us consider the language

$$L = \{\, w \in \{a, b\}^* \mid \forall u \in \{a, b\}^*, u \text{ is a prefix of } w : \#_a(u) \geq \#_b(u) \,\}.$$

The language L is clearly nonregular. However, let us consider a deterministic one-counter automaton $A = (K, \Sigma, \delta, q_0, F)$, defined as follows: $K = \{q_0\}$, $\Sigma = \{a, b\}$, $F = \{q_0\}$, and

$$\delta(q_0, a, 0) = (q_0, +1), \qquad \delta(q_0, a, 1) = (q_0, +1), \qquad \delta(q_0, b, 1) = (q_0, -1).$$

Clearly, $L(A) = L$. The strict state-$\mathcal{S}$-equiloadedness of the automaton A, for both $\mathcal{S}$ in $\{\mathcal{C}, \mathcal{A}\}$, follows from the fact that the automaton A has only one state. Thus, L is both in the family $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$, and in the family $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$.

Example 3.1.6 In this final example, we shall show that also the family $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$ contains some non-prefix-dense language. Let us consider the language

$$L = \{a^n b \mid n \geq 1\} \cup \{\varepsilon\}.$$

This language clearly is not prefix-dense. Let us consider the following deterministic one-counter automaton: $A = (K, \Sigma, \delta, q_0, F)$, where $K = \{q_0, q_1\}$, $\Sigma = \{a, b\}$, $F = \emptyset$, and

$$\begin{aligned} \delta(q_0, a, 0) &= (q_0, +1), \qquad & \delta(q_0, \varepsilon, 1) &= (q_1, 0), \\ \delta(q_1, a, 1) &= (q_0, 0), \qquad & \delta(q_1, b, 1) &= (q_1, -1). \end{aligned}$$

The statement $N(A) = L$, as well as the strict state-$\mathcal{S}$-equiloadedness of the automaton A for $\mathcal{S}$ in $\{\mathcal{C}, \mathcal{A}\}$, are considered to be clear. Thus, L is both in the family $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$, and in the family $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$.

3.2 Lemmas

In this section, we shall prove some auxiliary results that we shall use later on in our study of strictly $\mathcal{S}$-equiloaded deterministic one-counter automata.

Lemma 3.2.1 Let $A = (K, \Sigma, \delta, q_0, F)$ be a deterministic one-counter automaton.

1. Let an upper bound exist for the length of computation paths γ, such that γ reaches a configuration with the state q_0 and with the counter value 0 only at its beginning. Let H_1^* be the smallest of such upper bounds. Then,
$$H_1^* \leq |K|(|K| + 1).$$
2. Let an upper bound exist for the length of *accepting* computation paths γ, such that γ reaches a configuration with the state q_0 and with the counter value 0 only at its beginning. Let H_2^* be the smallest of such upper bounds. Then,
$$H_2^* \leq |K|(|K| + 1).$$
3. Let an upper bound exist for the length of computation paths γ *accepting by empty memory*, such that γ reaches a configuration with the state q_0 and with the counter value 0 only at its beginning. Let H_3^* be the smallest of such upper bounds. Then,
$$H_3^* \leq |K|(|K| + 1).$$

Proof. We shall prove all three claims at once. For the purpose of contradiction, let us suppose that $H_1^* > |K|(|K| + 1)$ (that $H_2^* > |K|(|K| + 1)$) [that $H_3^* > |K|(|K| + 1)$]. Then, a computation path γ_1 (an accepting computation path γ_2) [a computation path γ_3 accepting by empty memory] exists, such that a configuration with the state q_0 and with the counter value 0 is reached only at its beginning, and such that $|\gamma_1| = H_1^*$ ($|\gamma_2| = H_2^*$) [$|\gamma_3| = H_3^*$].

First, let us suppose that the greatest counter value achieved by the computation path γ_1 (the computation path γ_2) [the computation path γ_3] is $t \leq |K|$. Since, by our assumption, the length of the computation path is greater than $|K|(|K| + 1)$, it follows by the Pigeonhole principle that the computation path reaches at least two configurations with the same state and the same counter value. That is, the computation path γ_1 (the accepting computation path γ_2) [the computation path γ_3 accepting by empty memory] corresponds to some computation (some accepting computation) [some computation accepting by empty memory] of the form
$$(q_0, uvw, 0) \vdash^* (q, vw, t) \vdash^+ (q, w, t) \vdash^* (q', \varepsilon, t'),$$
where u, v, and w in Σ^* are words, t and t' in $\mathbb{N}$ are nonnegative integer counter values, and q and q' in K are states. Moreover, for the case of the computation path γ_2, the state q' is accepting (in F), and for the case of the computation path γ_3, the counter value t' is zero.

It is clear that there is a computation (an accepting computation) [a computation accepting by empty memory] on the word uv^2w of the form
$$(q_0, uv^2w, 0) \vdash^* (q, v^2w, t) \vdash^+ (q, vw, t) \vdash^+ (q, w, t) \vdash^* (q', \varepsilon, t').$$
Clearly, the length of the corresponding computation path (accepting computation path) [computation path accepting by empty memory] is greater than the length of the computation path γ_1 (the computation path γ_2) [the computation path γ_3]. However, this computation path clearly reaches a configuration with the state q_0 and with the counter value 0 only at its beginning, and thus, H_1^* (H_2^*) [H_3^*] is not an upper bound for the length of the corresponding set of computation paths. That is a contradiction.

Now, let us suppose that the greatest counter value achieved by the computation path γ_1 (the computation path γ_2) [the computation path γ_3] is $t \geq |K| + 1$. Then, it follows from the Pigeonhole principle that the computation path γ_1 (the computation path γ_2) [the computation path γ_3] corresponds to some computation (to some accepting computation) [to some computation accepting by empty memory] of the form

$$(q_0, uvwx, 0) \vdash^* (q, vwx, t_1) \vdash^+ (q, wx, t_2) \vdash^* (p, x, t) \vdash^* (p', \varepsilon, t'), \tag{3.1}$$

where u, v, w, and x are words in Σ^*, p, q, and p' in K are states, and t_1, t_2 and t' are nonnegative integer counter values in $\mathbb{N}$, such that $0 < t_1 < t_2$. Moreover, the counter value is always positive during the computation

$$(q, vwx, t_1) \vdash^+ (q, wx, t_2) \vdash^* (p, x, t).$$

For the case of the computation path γ_2, the state p' is in F, and for the case of the computation path γ_3, the counter value t' is zero.

Now, let us first suppose that the counter value is always positive during the computation

$$(p, x, t) \vdash^* (p', \varepsilon, t').$$

(Let us note that this may be the case only for the computation path γ_1 or for the computation path γ_2, not for the computation path γ_3.) It directly follows that

$$\begin{aligned}(q_0, uv^2wx, 0) &\vdash^* (q, v^2wx, t_1) \vdash^+ (q, vwx, t_2) \vdash^+ (q, wx, 2t_2 - t_1) \vdash^* \\ &\vdash^* (p, x, t + t_2 - t_1) \vdash^* (p', \varepsilon, t' + t_2 - t_1).\end{aligned}$$

The corresponding computation path clearly reaches the configuration with the state q_0 and with the counter value 0 only at its beginning. Moreover, it is longer than the computation path γ_1 (the computation path γ_2). Thus, H_1^* (H_2^*) is not an upper bound of the corresponding lengths of computation paths. That is a contradiction.

Let us now consider the remaining case when the counter value zero is reached during the computation

$$(p, x, t) \vdash^* (p', \varepsilon, t').$$

Then, it follows from the Pigeonhole principle that words x_1, x_2, x_3 in Σ^* exist, such that $x = x_1x_2x_3$, and such that the computation (3.1) can be rewritten as

$$\begin{aligned}(q_0, uvwx_1x_2x_3, 0) &\vdash^* (q, vwx_1x_2x_3, t_1) \vdash^+ (q, wx_1x_2x_3, t_2) \vdash^* (p, x_1x_2x_3, t) \vdash^* \\ &\vdash^* (q', x_2x_3, t_2') \vdash^+ (q', x_3, t_1') \vdash^* (p', \varepsilon, t'),\end{aligned}$$

for some state q' in K and some nonnegative integer counter values t_1', t_2' in $\mathbb{N}$, such that $0 < t_1' < t_2'$, so that the counter value is always positive during the computation

$$(p, x_1x_2x_3, t) \vdash^* (q', x_2x_3, t_2') \vdash^+ (q', x_3, t_1').$$

Now, let us denote by L the least common multiple of positive integers $t_2 - t_1$ and $t_2' - t_1'$. Let r, s in $\mathbb{N}^+$ be positive integers, such that $(r-1)(t_2 - t_1) = (s-1)(t_2' - t_1') = L$. Then, it is obvious that

$$\begin{aligned}(q_0, uv^rwx_1x_2^sx_3, 0) &\vdash^* (q, v^rwx_1x_2^sx_3, t_1) \vdash^+ (q, v^{r-1}wx_1x_2^sx_3, t_2) \vdash^+ \ldots \vdash^+ \\ &\vdash^+ (q, vwx_1x_2^sx_3, t_2 + (r-2)(t_2 - t_1)) \vdash^+ (q, wx_1x_2^sx_3, t_2 + (r-1)(t_2 - t_1)) \vdash^* \\ &\vdash^* (p, x_1x_2^sx_3, t + (r-1)(t_2 - t_1)) \vdash^* (q', x_2^sx_3, t_2' + (r-1)(t_2 - t_1)) \vdash^+ \\ &\vdash^+ (q', x_2^{s-1}x_3, t_1' + (r-1)(t_2 - t_1)) \vdash^+ \ldots \vdash^+ \\ &\vdash^+ (q', x_2x_3, t_1' + (r-1)(t_2 - t_1) - (s-2)(t_2' - t_1')) \vdash^+ \\ &\vdash^+ (q', x_3, t_1' + (r-1)(t_2 - t_1) - (s-1)(t_2' - t_1')) \vdash^0 (q', x_3, t_1') \vdash^* (p', \varepsilon, t').\end{aligned}$$

That is, $r-1$ loops from the state q, each increasing the counter value by $t_2 - t_1$, and $s-1$ loops from the state q', each decreasing the counter value by $t'_2 - t'_1$, can be added to the computation, without changing any of its important properties.

Clearly, the computation path corresponding to this computation reaches a configuration with the state q_0 and with the counter value 0 only at its beginning. Moreover, it is longer than the computation path γ_1 (the computation path γ_2) [the computation path γ_3].

Thus, H_1^* (H_2^*) [H_3^*] is not an upper bound of the corresponding lengths of computation paths. Once again, this is a contradiction. The lemma is proved. □

Lemma 3.2.2 Let $A = (K, \Sigma, \delta, q_0, F)$ be a deterministic one-counter automaton. Let q, q' in K be states and t in $\mathbb{N}$ be a nonnegative integer counter value. If the state q' is reachable from some configuration with the state q and with the counter value t, then a computation

$$(q, w, t) \vdash^* (q', \varepsilon, t')$$

exists for some w in Σ^* and t' in $\mathbb{N}$, such that the counter value does not exceed the value

$$T = t + |K|^3$$

during the computation.[1]

Proof. The lemma can be immediately seen to be true for $|K| = 1$. Thus, let us suppose that $|K| \geq 2$.

Let us denote by t_{min} the *minimal* nonnegative integer counter value, such that the state q' is reachable from some configuration with the state q and the counter value t, with the counter value not exceeding t_{min} during the corresponding computation. For the purpose of contradiction, let us suppose that $t_{min} > T$.

Clearly, since $|K|^3 \geq |K|^2$ and $t_{min} \geq t + |K|^3 + 1$, the computation corresponding to t_{min} is of the form

$$\begin{aligned}(q, u_1v_1x_1y_1u_2v_2x_2y_2u_3 \ldots u_kv_kx_kz, t) &\vdash^* (p_1^{(1)}, v_1x_1y_1u_2v_2x_2y_2u_3 \ldots u_kv_kx_kz, t + |K|^2 + 1) \vdash^* \\ &\vdash^* (p_1^{(2)}, x_1y_1u_2v_2x_2y_2u_3 \ldots u_kv_kx_kz, t_{min}) \vdash^* \\ &\vdash^* (p_1^{(3)}, y_1u_2v_2x_2y_2u_3 \ldots u_kv_kx_kz, t_{min}) \vdash^* \\ &\vdash^* (p_1^{(4)}, u_2v_2x_2y_2u_3 \ldots u_kv_kx_kz, t + |K|^2 + 1) \vdash^* \\ &\vdash^* (p_2^{(1)}, v_2x_2y_2u_3 \ldots u_kv_kx_kz, t + |K|^2 + 1) \vdash^* \\ &\vdash^* (p_2^{(2)}, x_2y_2u_3 \ldots u_kv_kx_kz, t_{min}) \vdash^* \\ &\vdash^* (p_2^{(3)}, y_2u_3 \ldots u_kv_kx_kz, t_{min}) \vdash^* \\ &\vdash^* (p_2^{(4)}, u_3 \ldots u_kv_kx_kz, t_{min}) \vdash^* \ldots \vdash^* \\ &\vdash^* (p_k^{(1)}, v_kx_kz, t + |K|^2 + 1) \vdash^* (p_k^{(2)}, x_kz, t_{min}) \vdash^* \\ &\vdash^* (p_k^{(3)}, z, t_{min}) \vdash^* (q', \varepsilon, t'). \end{aligned} \tag{3.2}$$

where u_i, v_i, x_i, $i = 1, \ldots, k$, y_i, $i = 1, \ldots, k-1$, and z in Σ^* are words, and where $p_i^{(1)}, p_1^{(2)}, p_i^{(3)}$, $i = 1, \ldots, k$, and $p_i^{(4)}$, $i = 1, \ldots, k-1$, are states. Moreover,

$$(p_i^{(2)}, x_iy_iu_{i+1} \ldots u_kv_kx_kz, t_{min})$$

is the *first* and

$$(p_i^{(3)}, y_iu_{i+1} \ldots u_kv_kx_kz, t_{min})$$

[1] This upper bound is not meant to be tight.

is the *last* configuration (without loss of generality, we assume that configurations do not repeat in the computation) in the computation

$$(p_i^{(1)}, v_i x_i y_i u_{i+1} \dots u_k v_k x_k z, t + |K|^2 + 1) \vdash^* (p_i^{(4)}, u_{i+1} \dots u_k v_k x_k z, t + |K|^2 + 1),$$

such that the counter value is t_{min}, for $i = 1, \dots, k-1$, and

$$(p_k^{(2)}, x_k z, t_{min})$$

is the *first* and

$$(p_k^{(3)}, z, t_{min})$$

is the *last* configuration in the computation

$$(p_k^{(1)}, v_k x_k z, t + |K|^2 + 1) \vdash^* (q', \varepsilon, t'),$$

such that the counter value is t_{min}. Furthermore, the counter value is *strictly less*[2] *than* t_{min} in computations

$$(q, u_1 v_1 x_1 y_1 u_2 v_2 x_2 y_2 u_3 \dots u_k v_k x_k z, t) \vdash^* (p_1^{(1)}, v_1 x_1 y_1 u_2 v_2 x_2 y_2 u_3 \dots u_k v_k x_k z, t + |K|^2 + 1)$$

and

$$(p_i^{(4)}, u_{i+1} v_{i+1} \dots u_k v_k x_k z, t + |K|^2 + 1) \vdash^* (p_{i+1}^{(1)}, v_{i+1} \dots u_k v_k x_k z, t + |K|^2 + 1)$$

for $i = 1, \dots, k-1$, and the counter value is *greater than or equal to* $t + |K|^2 + 1$ in computations

$$(p_i^{(1)}, v_i x_i y_i u_{i+1} \dots u_k v_k x_k z, t + |K|^2 + 1) \vdash^* (p_i^{(4)}, u_{i+1} \dots u_k v_k x_k z, t + |K|^2 + 1),$$

for $i = 1, \dots, k-1$, and

$$(p_k^{(1)}, v_k x_k z, t + |K|^2 + 1) \vdash^* (p_k^{(3)}, z, t_{min}).$$

That is, the configuration can be divided into parts corresponding to the subwords of the input word as follows: in its parts corresponding to words u_i, $i = 1, \dots, k$, the computation never reaches the counter value t_{min}, i.e., the counter value is strictly less than t_{min}. That is, to obtain the contradiction (i.e., to construct a computation reaching the state q' from some configuration with the state q and the counter value t, such that the counter value is strictly less than t_{min} during the computation), nothing has to be done with these parts. In its parts corresponding to the words $v_i x_i y_i$, $i = 1, \dots, k-1$, the counter value is always greater than or equal to $t + |K|^2 + 1$. In the process of constructing a computation, existence of which implies the contradiction, this fact shall enable us to omit some computation steps in these parts of the computation, without reaching the counter value 0. The same fact is true for the part of the computation corresponding to the word $v_k x_k$. Moreover, the counter value t_{min} is reached for the first time in these parts after reading the subword v_i, and for the last time after reading the subword x_i.

Now, it follows from the Pigeonhole principle that if t_1, t_2 in $\mathbb{N}$ are nonnegative integer counter values, such that $t_2 - t_1 \geq |K|$, then every computation

$$(q_1, y, t_1) \vdash^* (q_2, \varepsilon, t_2),$$

where y in Σ^* is a word and q_1, q_2 in K are states, is of the form

$$(q_1, y_1 y_2 y_3, t_1) \vdash^* (q_3, y_2 y_3, t_3) \vdash^* (q_3, y_3, t_4) \vdash^* (q_2, \varepsilon, t_2),$$

where y_1, y_2, y_3 are words in Σ^*, such that $y = y_1 y_2 y_3$, and t_3, t_4 in $\mathbb{N}$ are nonnegative integer counter values, $t_1 \leq t_3 \leq t_4 \leq t_2$, such that $t_4 - t_3$ is in $\{1, 2, \dots, |K|\}$.

[2]This assumption can be made since $|K| \geq 2$.

Similarly, if t'_1, t'_2 in $\mathbb{N}$ are nonnegative integer counter values, such that $t'_1 - t'_2 \geq |K|$, then every computation
$$(q'_1, y', t'_1) \vdash^* (q'_2, \varepsilon, t'_2),$$
where y' in Σ^* is a word and q'_1, q'_2 in K are states, is of the form
$$(q'_1, y'_1y'_2y'_3, t'_1) \vdash^* (q'_3, y'_2y'_3, t'_3) \vdash^* (q'_3, y'_3, t'_4) \vdash^* (q'_2, \varepsilon, t'_2),$$
where y'_1, y'_2, y'_3 in Σ^* are words, such that $y' = y'_1y'_2y'_3$, and t'_3, t'_4 in $\mathbb{N}$ are nonnegative integer counter values, $t'_1 \geq t'_3 \geq t'_4 \geq t'_2$, such that $t'_3 - t'_4$ is in $\{1, 2, \ldots, |K|\}$.

We shall call computations of the form
$$(p, u, t_1) \vdash^* (p, \varepsilon, t_2),$$
where u in Σ^* is a word, p in K is a state, and t_1, t_2 in $\mathbb{N}$ are nonnegative integer counter values, such that $1 \leq t_2 - t_1 \leq |K|$ resp. $1 \leq t_1 - t_2 \leq |K|$, *bounded increasing* resp. *bounded decreasing loops*. The property observed above can be thus restated as that every computation that increases the counter by at least $|K|$ contains a bounded increasing loop, and every computation that decreases the counter by at least $|K|$ contains a bounded decreasing loop.

Now, let us return to the computation (3.2). Since, for $i = 1, \ldots, k$, the counter value is increased by the value of
$$t_{min} - (t + |K|^2 + 1) \geq t + |K|^3 + 1 - t - |K|^2 - 1 = |K|^3 - |K|^2$$
in the computation
$$(p_i^{(1)}, v_ix_iy_iu_{i+1} \ldots u_kv_kx_kz, t + |K|^2 + 1) \vdash^* (p_i^{(2)}, x_iy_iu_{i+1} \ldots u_kv_kx_kz, t_{min}), \tag{3.3}$$
it follows from the Generalized pigeonhole principle that the computation (3.3) contains at least $|K|$ nonoverlapping[3] bounded increasing loops, increasing the counter by the same value t_i^+ in $\{1, 2, \ldots, |K|\}$.

Similarly, since for $i = 1, \ldots, k-1$, the counter value is decreased by the value of
$$t_{min} - (t + |K|^2 + 1) \geq t + |K|^3 + 1 - t - |K|^2 - 1 = |K|^3 - |K|^2$$
in the computation
$$(p_i^{(3)}, y_iu_{i+1} \ldots u_kv_kx_kz, t_{min}) \vdash^* (p_i^{(4)}, u_{i+1} \ldots u_kv_kx_kz, t + |K|^2 + 1), \tag{3.4}$$
the computation (3.4) contains at least $|K|$ nonoverlapping bounded decreasing loops, decreasing the counter by the same value t_i^- in $\{1, 2, \ldots, |K|\}$.

Now, it can be easily seen that, for $i = 1, \ldots, k-1$, t_i^- of these bounded *increasing* loops and t_i^+ of these bounded *decreasing* loops can be deleted, without changing almost any important property of the computation. In fact, the computation
$$\begin{aligned}(p_i^{(1)}, v_ix_iy_iu_{i+1} \ldots u_kv_kx_kz, t + |K|^2 + 1) &\vdash^* (p_i^{(2)}, x_iy_iu_{i+1} \ldots u_kv_kx_kz, t_{min}) \vdash^* \\ &\vdash^* (p_i^{(3)}, y_iu_{i+1} \ldots u_kv_kx_kz, t_{min}) \vdash^* \\ &\vdash^* (p_i^{(4)}, u_{i+1} \ldots u_kv_kx_kz, t + |K|^2 + 1)\end{aligned}$$
becomes
$$\begin{aligned}(p_i^{(1)}, v'_ix'_iy'_iu_{i+1} \ldots u_kv_kx_kz, t + |K|^2 + 1) &\vdash^* (p_i^{(2)}, x'_iy'_iu_{i+1} \ldots u_kv_kx_kz, t_{min} - t_i^+ \cdot t_i^-) \vdash^* \\ &\vdash^* (p_i^{(3)}, y'_iu_{i+1} \ldots u_kv_kx_kz, t_{min} - t_i^+ \cdot t_i^-) \vdash^* \\ &\vdash^* (p_i^{(4)}, u_{i+1} \ldots u_kv_kx_kz, t + |K|^2 + 1)\end{aligned}$$

[3] In the sense that there are not any two of these bounded increasing loops with common computation step.

for some words v'_i, x'_i, y'_i in Σ^*. This property holds, since the counter value 0 is never reached in this computation (since $t_i^+ \cdot t_i^- \le |K|^2 < t + |K|^2 + 1$). However, the greatest counter value reached by this computation is $t_{min} - t_i^+ \cdot t_i^- < t_{min}$.

That is, to construct a computation reaching the state q' from some configuration with the state q and the counter value t (existence of which would lead to a contradiction), it suffices to do analogously decrease the greatest counter value reached in the part of the computation corresponding to the input subword $u_k v_k x_k$. However, this can be done easily: if in the computation

$$(p_k^{(3)}, z, t_{min}) \vdash^* (q', \varepsilon, t'),$$

the counter value $t + |K|^2 + 1$ is reached, then the bounded increasing loops and bounded decreasing loops can be deleted from the computation in exactly the same manner as above. Otherwise, it suffices to simply delete one bounded increasing loop. That is, the statement of the lemma is proved. □

Lemma 3.2.3 Let $A = (K, \Sigma, \delta, q_0, F)$ be a deterministic one-counter automaton. Let p, q in K be states, and s, t in $\mathbb{N}$ be nonnegative integer counter values. If, for some u in Σ^*, $(p, u, s) \vdash^* (q, \varepsilon, t)$, then a nonnegative integer n in $\mathbb{N}$ exists, such that

$$n \le \max\{s, t\} \cdot |K| + |K|^4 + |K| - 1,$$

and $(p, v, s) \vdash^n (q, \varepsilon, t)$ for some word v in Σ^*.

Proof. Let n be the *smallest* nonnegative integer, such that, for some v in Σ^*,

$$(p, v, s) \vdash^n (q, \varepsilon, t). \tag{3.5}$$

We shall prove that $n \le \max\{s, t\} \cdot |K| + |K|^4 + |K| - 1$ by showing that the counter value is always less than or equal to $\max\{s, t\} + |K|^3$ during the computation (3.5).

For the purpose of contradiction, let us suppose that this is not true, i.e., that the computation (3.5) can be rewritten as

$$\begin{aligned}(p, v_1v_2v_3v_4, s) &\vdash^{n_1} (q_1, v_2v_3v_4, \max\{s,t\} + |K|^2 + 1) \vdash^{n_2} (q_2, v_3v_4, \max\{s,t\} + |K|^3 + 1) \vdash^{n_3} \\ &\vdash^{n_3} (q_3, v_4, \max\{s,t\} + |K|^2 + 1) \vdash^{n_4} (q, \varepsilon, t)\end{aligned}$$

for some words v_1, v_2, v_3, v_4 in Σ^*, such that $v_1v_2v_3v_4 = v$, and nonnegative integers n_1, n_2, n_3, n_4 in $\mathbb{N}$, such that $n_1 + n_2 + n_3 + n_4 = n$, so that the counter value is always greater than or equal to $\max\{s, t\} + |K|^2 + 1$ during the computation

$$\begin{aligned}(q_1, v_2v_3v_4, \max\{s,t\} + |K|^2 + 1) &\vdash^{n_2} (q_2, v_3v_4, \max\{s,t\} + |K|^3 + 1) \vdash^{n_3} \\ &\vdash^{n_3} (q_3, v_4, \max\{s,t\} + |K|^2 + 1).\end{aligned}$$

In the computation

$$(q_1, v_2v_3v_4, \max\{s,t\} + |K|^2 + 1) \vdash^{n_2} (q_2, v_3v_4, \max\{s,t\} + |K|^3 + 1), \tag{3.6}$$

the counter value is increased by $|K|^3 - |K|^2$, and is decreased by the same value in the computation

$$\vdash^{n_2} (q_2, v_3v_4, \max\{s,t\} + |K|^3 + 1) \vdash^{n_3} (q_3, v_4, \max\{s,t\} + |K|^2 + 1). \tag{3.7}$$

Thus, it follows from the Generalized pigeonhole principle that at least $|K|$ nonoverlapping bounded increasing loops[4] exist in the computation (3.6), each increasing the counter value by the same value of t^+ in $\mathbb{N}$. Similarly, the computation (3.7) contains at least $|K|$ nonoverlapping bounded decreasing loops, each decreasing the counter value by some t^- in $\mathbb{N}$. Since the counter

[4]For the definition, see the proof of Lemma 3.2.2.

value is greater than or equal to $\max\{s,t\} + |K|^2 + 1$ during both of the computations, t^- such increasing loops and t^+ such decreasing loops can be deleted.

The resulting computation clearly begins in some configuration with the state p and the counter value s, and ends in some configuration with the state q and the counter value t. However, it is shorter than the original computation, and that contradicts the assumption that n is the smallest nonnegative integer with the above specified properties.

Thus, we have proved that the counter value is always less than or equal to $\max\{s,t\} + |K|^3$ during the computation (3.5). Now, we shall prove that

$$n \leq \max\{s,t\} \cdot |K| + |K|^4 + |K| - 1.$$

For the purpose of contradiction, let us suppose that

$$n \geq \max\{s,t\} \cdot |K| + |K|^4 + |K|.$$

Since there are exactly $\max\{s,t\} + |K|^3 + 1$ possibilities for the counter value in the computation (3.5), it follows by the Pigeonhole principle that the computation (3.5) has a form

$$(p, w_1w_2w_3, s) \vdash^* (q_{rep}, w_2w_3, t_{rep}) \vdash^+ (q_{rep}, w_3, t_{rep}) \vdash^* (q, \varepsilon, t),$$

for some words w_1, w_2, w_3 in Σ^*, such that $w_1w_2w_3 = v$, for some state q_{rep} in K, and for some nonnegative integer counter value t_{rep} in $\mathbb{N}$. However, the part

$$(q_{rep}, w_2w_3, t_{rep}) \vdash^+ (q_{rep}, w_3, t_{rep})$$

of the computation can be omitted, and this results in a shorter computation beginning in a configuration with the state p and the counter value s, and ending in a configuration with the state q and the counter value t. This contradicts the minimality of n. □

Corollary 3.2.4 Let $A = (K, \Sigma, \delta, q_0, F)$ be a deterministic one-counter automaton, let q in K be a state, and t in $\mathbb{N}$ be a nonnegative integer counter value.

1. If some configuration with the state q_0 and the counter value 0 is reachable from some configuration with the state q and the counter value t, then it is reachable in at most

$$(t+1) \cdot |K| + |K|^4 - 1$$

computation steps.

2. If some accepting configuration is reachable from some configuration with the state q and the counter value t, then it is reachable in at most

$$(t+1) \cdot |K| + 2 \cdot |K|^4 - 1$$

computation steps.

3. If some configuration with the counter value 0 is reachable from some configuration with the state q and the counter value t, then it is reachable in at most

$$(t+1) \cdot |K| + |K|^4 - 1$$

computation steps.

Proof. The claims 1 and 3 are direct corollaries of Lemma 3.2.3. The claim 2 follows from Lemma 3.2.2 and Lemma 3.2.3. □

3.3 Characterization of Strict Transition-$\mathcal{S}$-Equiloadedness

In this section, we shall prove the characterization of strictly transition-$\mathcal{S}$-equiloaded deterministic one-counter automata, for $\mathcal{S} = \mathcal{C}$, $\mathcal{S} = \mathcal{A}$, and $\mathcal{S} = \mathcal{E}$. We shall prove this characterization in Theorem 3.3.1. Later in this section, we shall prove that this characterization is decidable and present an algorithm for deciding if a given deterministic one-counter automaton is strictly transition-$\mathcal{S}$-equiloaded.

Theorem 3.3.1 Let $A = (K, \Sigma, \delta, q_0, F)$ be a deterministic one-counter automaton.

a) A is strictly transition-$\mathcal{C}$-equiloaded, if and only if the following two properties hold:

(i) For every computation path γ, such that $|\gamma| \geq 1$ and γ reaches a configuration with the state q_0 and the counter value 0 at its beginning, at its end, but not otherwise, the property
$$\#[e, \gamma] = 1$$
holds for every transition e in D.

(ii) If γ is a computation path of the automaton A, such that γ reaches a configuration with the state q_0 and the counter value 0 only at its beginning, then $|\gamma| \leq M$ for some fixed constant M in $\mathbb{N}$.

b) A is strictly transition-$\mathcal{A}$-equiloaded, if and only if $L(A)$ is empty or if the following two properties hold:

(i) For every computation path γ, such that $|\gamma| \geq 1$ and γ reaches a configuration with the state q_0 and the counter value 0 at its beginning, at its end, but not otherwise, the property
$$\#[e, \gamma] = 1$$
holds for every transition e in D.

(ii') If γ is an accepting computation path of the automaton A, such that γ reaches a configuration with the state q_0 and the counter value 0 only at its beginning, then $|\gamma| \leq M$ for some fixed constant M in $\mathbb{N}$.

c) A is strictly transition-$\mathcal{E}$-equiloaded, if and only if the following two properties hold:

(i) For every computation path γ, such that $|\gamma| \geq 1$ and γ reaches a configuration with the state q_0 and the counter value 0 at its beginning, at its end, but not otherwise, the property
$$\#[e, \gamma] = 1$$
holds for every transition e in D.

(ii'') If γ is a computation path of the automaton A accepting by empty memory, such that γ reaches a configuration with the state q_0 and the counter value 0 only at its beginning, then $|\gamma| \leq M$ for some fixed constant M in $\mathbb{N}$.

Proof. We shall prove all three claims at once. First, let us prove the left-to-right implications. For the purpose of contradiction, let us suppose that the automaton A is strictly transition-$\mathcal{S}$-equiloaded for some $\mathcal{S}$ in $\{\mathcal{C}, \mathcal{A}, \mathcal{E}\}$, and that the property (i) does not hold. Moreover, if $\mathcal{S} = \mathcal{A}$, let us further suppose that $L(A)$ is nonempty.

Since (i) does not hold, a computation path γ exists, such that γ corresponds to a computation
$$(q_0, w, 0) \vdash^n (q_0, \varepsilon, 0)$$
for some word w in Σ^* and some positive integer n in $\mathbb{N}^+$, and
$$|\#[e, \gamma] - \#[f, \gamma]| \geq 1$$

for some e, f in D (this is the case because every transition leading from the state q_0 at the counter value 0 can be used at most once in computation paths that we are concerned with in (i)). That is, for all k in $\mathbb{N}$, a computation path γ_k exists, such that γ_k corresponds to a computation

$$(q_0, w^k, 0) \vdash^{kn} (q_0, \varepsilon, 0),$$

and

$$|\#[e, \gamma_k] - \#[f, \gamma_k]| \geq k. \tag{3.8}$$

In the case $\mathcal{S} = \mathcal{C}$, (3.9) clearly contradicts an assumption of strict transition-$\mathcal{C}$-equiloadedness of the automaton A. Moreover, γ_k is clearly a computation path accepting by empty memory for all k in $\mathbb{N}$. Thus, (3.9) clearly leads to a contradiction also in the case $\mathcal{S} = \mathcal{E}$. Finally, in the case $\mathcal{S} = \mathcal{A}$, there has to be a nonnegative integer *constant* r, such that

$$(q_0, u, 0) \vdash^{r} (q, \varepsilon, t)$$

for some word u in Σ^*, some *accepting* state q in F, and some nonnegative integer t in $\mathbb{N}$. Thus, for an *accepting* computation path γ'_k, corresponding to the computation

$$(q_0, w^k u, 0) \vdash^{kn} (q_0, u, 0) \vdash^{r} (q, \varepsilon, t),$$

the property

$$\left|\#[e, \gamma'_k] - \#[f, \gamma'_k]\right| \geq k - r \tag{3.9}$$

holds. This clearly contradicts the assumption that the automaton A is strictly transition-$\mathcal{A}$-equiloaded.

Now, let us suppose that $\mathcal{S} = \mathcal{C}$ and the property (ii) does not hold. Then, for each l in $\mathbb{N}$, a computation path κ_l exists, such that $|\kappa_l| \geq l$, and $\#[e, \kappa_l] \leq 1$ for each transition $e = (q_0, c, 0, q', r)$, where c is in Σ, q' is in K, and r is in $\{0, 1\}$. However, by the Pigeonhole principle, a transition f in D exists, such that

$$\#[f, \kappa_l] \geq \frac{l}{|D|}.$$

Thus, we may conclude that for all k in $\mathbb{N}$, a computation path κ_{l_k} exists, such that

$$\left|\#[e, \kappa_{l_k}] - \#[f, \kappa_{l_k}]\right| \geq k.$$

This clearly contradicts the assumption that the automaton A is strictly transition-$\mathcal{C}$-equiloaded.

For the cases $\mathcal{S} = \mathcal{A}$ and $\mathcal{S} = \mathcal{E}$, an analogous reasoning can be used to obtain a contradiction, if we suppose that the property (ii') resp. (ii'') does not hold.

Now, let us prove the easier right-to-left implications. We consider to be obvious that if the properties (i) and (ii) are satisfied, then the inequality

$$|\#[e, \gamma] - \#[f, \gamma]| \leq M \tag{3.10}$$

holds for all computation paths γ, and each two transitions e, f in D. Similarly, if (i) and (ii') are satisfied, the inequality (3.10) holds for all accepting computation paths γ, and if (i) and (ii'') are satisfied, the inequality holds for all computation paths γ accepting by empty memory. The theorem is proved. □

In what follows, we shall show that the previous theorem is a good characterization of strictly transition-$\mathcal{S}$-equiloaded DOCA. That is, we shall prove that the characterization from the previous theorem is decidable: we shall present an algorithm deciding the strict transition-$\mathcal{S}$-equiloadedness of a given DOCA by deciding, if the characterization from Theorem 3.3.1 is satisfied.

The algorithm shall proceed as follows: first, it examines all computation paths γ of length at most

$$M = \max\{|D|, |K|(|K| + 1)\} + 1,$$

such that γ visits a configuration with the state q_0 and the counter value 0 only at its beginning. This is done by generating all possible computation paths. If a computation path that visits a configuration with the state q_0 and the counter value 0 for the second time is discovered, it is thrown away. However, before this is done, the algorithm checks if each transition is used exactly once in the computation path. If not, the property (i) is violated, and the automaton is not strictly transition-$\mathcal{S}$-equiloaded.

At the end of this part of the algorithm, all computation paths γ of length exactly M are generated, such that γ visits a configuration with the state q_0 and the counter value 0 only at its beginning. It is obvious that the property (i) of the characterization from Theorem 3.3.1 is satisfied, if and only if none of these computation paths can be prolonged to a computation path ending in a configuration with the state q_0 and the counter value 0. Corollary 3.2.4, together with a simple fact that the counter value at the end of a computation path is always less than or equal to the length of the computation path, implies that to decide if this property holds, it suffices to check all possible prolongments by at most

$$(M+1)\cdot|K|+|K|^4-1$$

transitions. Moreover, it follows from Lemma 3.2.1 that the property (ii) of the characterization from Theorem 3.3.1 is satisfied, if and only if the set of computation paths generated by the first part of the algorithm is empty. Further, by Lemma 3.2.1, the property (ii') of the characterization is satisfied, if and only if none of these computation paths is accepting, and none of them can be prolonged to an accepting computation path. It follows from Corollary 3.2.4 that it suffices to check all possible prolongments by at most

$$(M+1)\cdot|K|+2\cdot|K|^4-1$$

transitions. Finally, again by Lemma 3.2.1, the property (ii'') of the characterization is satisfied, if and only if none of these computation paths is accepting by empty memory, and none of them can be prolonged to a computation path accepting by empty memory. Corollary 3.2.4 implies that it suffices to check all possible prolongments by at most

$$(M+1)\cdot|K|+|K|^4-1$$

transitions. These observations result in Algorithm 1.

Algorithm 1 Deciding a strict transition-$\mathcal{S}$-equiloadedness of a DOCA by checking if the characterization from Theorem 3.3.1 is satisfied.

Input: Finitely described DOCA $A=(K,\Sigma,\delta,q_0,F)$, name of the function $\mathcal{S}$ in $\{\mathcal{C},\mathcal{A},\mathcal{E}\}$.
Output: TRUE, if the automaton A is strictly transition-$\mathcal{S}$-equiloaded, FALSE otherwise.

$M \leftarrow \max\{|D|,|K|(|K|+1)\}+1$
$C \leftarrow \{$the empty computation path$\}$
for $i \leftarrow 1, M$ **do**
  $C' \leftarrow \emptyset$
  for each γ in C **do**
    Let S be the set of all computation paths γ' of length i, such that γ' is a computation path γ prolonged by one extra transition.
    $C' \leftarrow C' \cup S$
  end for
  $C \leftarrow C'$

Algorithm 1 continued.

for each γ in C **do**
if γ begins and ends in a configuration with the state q_0 and the counter value 0 **then**
$C \leftarrow C - \{\gamma\}$
if $\#[e,\gamma] \neq 1$ for some e in D **then**
return FALSE
end if
end if
end for
end for
if $\mathcal{S} = \mathcal{C}$ **then**
if $C = \emptyset$ **then**
return TRUE
else
return FALSE
end if
else if $\mathcal{S} = \mathcal{A}$ **then**
for $i \leftarrow 1, (M+1) \cdot |K| + 2 \cdot |K|^4$ **do**
for each γ in C **do**
if γ ends in an accepting state or in q_0 with the counter value 0 **then**
return FALSE
end if
end for
$C' \leftarrow \emptyset$
for each γ in C **do**
Let S be the set of all computation paths γ' of length i, such that γ' is a computation path γ prolonged by one extra transition.
$C' \leftarrow C' \cup S$
end for
$C \leftarrow C'$
end for
return TRUE
else if $\mathcal{S} = \mathcal{E}$ **then**
for $i \leftarrow 1, (M+1) \cdot |K| + |K|^4$ **do**
for each γ in C **do**
if γ ends in a configuration with the counter value 0 **then**
return FALSE
end if
end for
$C' \leftarrow \emptyset$
for each γ in C **do**
Let S be the set of all computation paths γ' of length i, such that γ' is a computation path γ prolonged by one extra transition.
$C' \leftarrow C' \cup S$
end for
$C \leftarrow C'$
end for
return TRUE
end if

3.4 Further Lemmas

In this section, we shall state and prove some additional lemmas that we shall use mainly in our study of relations between various families of strictly $\mathcal{S}$-equiloaded DOCA-languages.

Lemma 3.4.1 Let $A = (K, \Sigma, \delta, q_0, F)$ be a strictly transition-$\mathcal{S}$-equiloaded DOCA, for some $\mathcal{S}$ in $\{\mathcal{C}, \mathcal{A}, \mathcal{E}\}$, accepting an infinite language. Then, for each q in K, $\delta(q, c, 0)$ is defined for at most one c in $\Sigma \cup \{\varepsilon\}$.

Proof. By contradiction. Let us suppose that two distinct (and thus necessarily non-ε) symbols c, d in Σ exist, such that both $\delta(q, c, 0)$ and $\delta(q, d, 0)$ are defined for some state q in K. That is, $(q, c, 0, \mathrm{pr}_1(\delta(q, c, 0)), \mathrm{pr}_2(\delta(q, c, 0)))$ and $(q, d, 0, \mathrm{pr}_1(\delta(q, d, 0)), \mathrm{pr}_2(\delta(q, d, 0)))$ are two distinct transitions in D.

Since the language $L(A)$ is infinite, it follows from the characterization given in Theorem 3.3.1 that

$$(q_0, w, 0) \vdash^+ (q_0, \varepsilon, 0),$$

for some word w in Σ^*. Moreover, the property (i) of the characterization from Theorem 3.3.1 has to hold. That is, for every computation path γ, such that $|\gamma| \geq 1$ and γ reaches a configuration with the state q_0 and the counter value 0 at its beginning, at its end, but not otherwise, the property

$$\#[e, \gamma] = 1$$

holds for every transition e in D. Now, if $q = q_0$, this is a clear contradiction. Let us therefore suppose that $q \neq q_0$. Then, since there are at least two distinct transitions leading from q at the counter value 0, every such computation path γ has to visit the state q at least twice with the counter value 0. Now, let us consider γ to be fixed. Without loss of generality, let us suppose that the transition $(q, c, 0, \mathrm{pr}_1(\delta(q, c, 0)), \mathrm{pr}_2(\delta(q, c, 0)))$ is used before the transition $(q, d, 0, \mathrm{pr}_1(\delta(q, d, 0)), \mathrm{pr}_2(\delta(q, d, 0)))$ in the computation path γ. Then, obviously, there is a computation subpath γ' of γ, corresponding to some computation

$$(q, cu, 0) \vdash^+ (q, \varepsilon, 0)$$

for some u in Σ^*, such that a configuration with the state q_0 and with the counter value 0 is not visited by γ'. Clearly, γ' can be iterated arbitrarily many times as a subpath of γ. Thus, a computation path violating the property (i) exists, and that is a contradiction. □

Lemma 3.4.2 Let $A = (K, \Sigma, \delta, q_0, F)$ be a strictly transition-$\mathcal{S}$-equiloaded DOCA, for some $\mathcal{S}$ in $\{\mathcal{C}, \mathcal{A}, \mathcal{E}\}$, accepting an infinite language. Then, a unique positive integer k in $\mathbb{N}^+$, a unique sequence of states $\{q_i\}_{i=1}^{k+1}$ in K^{k+1}, and a unique partition of the set of transitions $\{S_i\}_{i=1}^{k}$ in $\left(2^D\right)^k$ exists, such that the following statement holds: let γ be an arbitrary computation path of the automaton A, such that $|\gamma| \geq 1$ and γ reaches a configuration with the state q_0 and the counter value 0 at its beginning, at its end, but not otherwise. Then, the sequence of states of the automaton A visited by γ with the counter value being 0 is identical to $\{q_i\}_{i=1}^{k+1}$, and the set of transitions used by γ between the i-th and $(i+1)$-th visit of a configuration with the counter value 0 is S_i, for $i = 1, \ldots, k$. Clearly, $q_1 = q_{k+1} = q_0$.

Proof. We shall first prove that k is unique. This can be easily observed since, according to Lemma 3.4.1, every computation path γ satisfying the imposed conditions has to visit each state q in K, such that $\delta(q, c, 0)$ is defined for some c in $\Sigma \cup \{\varepsilon\}$, exactly once (except $q = q_0$ that is visited twice). Clearly, no other state can be visited by γ with the counter value being 0, since the computation would get stuck.

Now, we shall prove that $\{q_i\}_{i=1}^{k+1}$ is unique. For the purpose of contradiction, let us suppose that $\{p_i\}_{i=1}^{k+1}$ is a sequence of states, such that it is followed (in the sense of the statement of the lemma) by some computation path γ_1, and $\{r_i\}_{i=1}^{k+1}$ is a different sequence of states, such that it is followed by some different computation path γ_2, where both computation paths γ_1, γ_2 satisfy the conditions imposed in the statement of the lemma. By what we have stated above, the second sequence can be viewed as a nonidentical permutation of the first sequence (however, this permutation is required to preserve the first and the last element). Thus, integers i, j in $\{1, \ldots, k\}$ exist, such that $i < j$, and $p_i = r_j$. Clearly, the computation path γ_1 can be cut after reaching a configuration with the state p_i and the counter value 0, and then continued as the computation path γ_2. However, the resulting computation path γ clearly contradicts the property that k is always unique, since the number of visits of a configuration with the counter value 0 by the computation path γ is strictly less than $k+1$.

Finally, we shall prove that $\{S_i\}_{i=1}^{k}$ is unique. Since the automaton A is strictly transition-$\mathcal{S}$-equiloaded for some $\mathcal{S}$ in $\{\mathcal{C}, \mathcal{A}, \mathcal{E}\}$, and since the accepted language $L(A)$ is infinite, it follows from the characterization given in Theorem 3.3.1 that a computation path γ exists, such that $|\gamma| \geq 1$ and γ reaches a configuration with the state q_0 and the counter value 0 at its beginning, at its end, but not otherwise. By the property (i) of the characterization from Theorem 3.3.1, the property $\#[e, \gamma] = 1$ has to hold for all e in D.

Let us suppose that $\{T_i\}_{i=1}^{k}$ is a sequence of sets of transitions, followed by (in the sense of the statement of the lemma) the computation path γ. By our assumption, at least one another sequence of sets of transitions $\{U_i\}_{i=1}^{k}$ exists, such that it is followed by some another computation path, say γ', such that it is nonempty, and such that it reaches a configuration with the state q_0 and the counter value 0 at its beginning, at its end, but not otherwise. Then, at least one index j in $\{1, \ldots, k\}$ exists, such that $T_j \neq U_j$. Moreover, since the sequence $\{q_i\}_{i=1}^{k+1}$ is unique, it follows that the computation of the form $(q_j, u, 0) \vdash^* (q_{j+1}, \varepsilon, 0)$ exists for some u in Σ^*, such that it uses exactly all transitions from T_j (as in the computation path γ), and another computation of the form $(q_j, v, 0) \vdash^* (q_{j+1}, \varepsilon, 0)$ exists for some v in Σ^*, such that it uses exactly all transitions from U_j (as in the computation path γ').

Now, let us consider a computation path γ'' such that it is identical to γ except that between the states q_j and q_{j+1} it follows the path of γ', i.e., the corresponding computation is the above mentioned computation of the form $(q_j, v, 0) \vdash^* (q_{j+1}, \varepsilon, 0)$. Clearly, $|\gamma''| \geq 1$ and γ'' reaches a configuration with the state q_0 and the counter value 0 at its beginning, at its end, but not otherwise. However, it is obvious that the property $\#[e, \gamma] = 1$ cannot hold for all e in D. This contradicts the characterization of strictly transition-equiloaded DOCA given in Theorem 3.3.1. The lemma is proved. $\square$

3.5 Families of Strictly $\mathcal{S}$-Equiloaded Languages

In this section, we shall examine the relations that hold between various families of strictly $\mathcal{S}$-equiloaded DOCA-languages and the relations of these families to some families of languages studied earlier in this report.

Theorem 3.5.1 The following strict inclusions hold:

1. $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C}) \subsetneq \mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C})$, 2. $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E}) \subsetneq \mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$.

Proof. We shall prove only the first statement, the proof of the second statement is analogous. Let L in $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$ be a strictly transition-$\mathcal{C}$-equiloaded DOCA-language accepted by empty memory. Let $A = (K, \Sigma, \delta, q_0, F)$ be a strictly transition-$\mathcal{C}$-equiloaded deterministic one-counter automaton, such that $N(A) = L$. We shall construct a strictly transition-$\mathcal{C}$-equiloaded deterministic one-counter automaton $A' = (K', \Sigma', \delta', q_0', F')$, such that $L(A') = L$, as follows: $K' = K \times \{old, new\}$, $\Sigma' = \Sigma$, the transition function δ' is defined for all q, q' in K, c in $\Sigma' \cup \{\varepsilon\}$

and r in $\{-1,0,1\}$ by

$$\begin{aligned}
\delta'((q,old),c,1) &= ((q',old),r) \iff \delta(q,c,1) = (q',r),\\
\delta'((q,old),\varepsilon,0) &= ((q,new),0),\\
\delta'((q,new),c,0) &= ((q',old),r) \iff \delta(q,c,0) = (q',r),
\end{aligned}$$

$q'_0 = (q_0, old)$, and $F' = K \times \{new\}$. The idea behind this construction is to replace each state of the automaton A by two states: the „old" one and the „new" one. If the counter value is greater than zero, then the computation proceeds by using old states, exactly as in the automaton A. However, when the counter value 0 is reached, the computation first makes an ε-transition from the old state to the new states and only then continues to simulate the computation of the automaton A (by using old states, again). The reason why the new states are introduced is that a state (q, new) of the automaton A' can be reached after reading some w in Σ^* if and only if some configuration of the automaton A with the state q and the counter value 0 can be reached after reading w. Thus, if A accepts w by empty memory, then some new state of the automaton A' can be reached after reading w. Conversely, if some new state of A' can be reached after reading w, then A accepts w by empty memory. It therefore suffices to define the set F' to be $K \times \{new\}$.

Conversely, every language in $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$ has to contain the empty word ε. However, it is easy to find a language L in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C})$, such that ε is not in L. The language $L = \{a\}^+$ may serve as an easy counterexample. The construction of a strictly transition-$\mathcal{C}$-equiloaded DOCA accepting L is easy and left to the reader. The theorem is proved. $\square$

Theorem 3.5.2 The following strict inclusions hold:

1. $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C}) \subsetneq \mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$,
2. $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C}) \subsetneq \mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E})$.

Proof. The improper inclusions follow immediately from Theorem 1.6.3. We shall prove that these inclusions are proper.

Let us consider the language $L = \{abcd, acbd, bacd, bcad\}^*$. We shall prove that this language is in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$, but not in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C})$.

We shall construct a deterministic one-counter automaton $A = (K, \Sigma, \delta, q_0, F)$, such that A is strictly transition-$\mathcal{A}$-equiloaded, and such that $L(A) = L$ as follows: the set of states K shall be defined by

$$K = \{q_i \mid i = 0, \dots, 21\} \cup \{p_{1,i} \mid i = 1, \dots, 4\} \cup \{p_{2,i} \mid i = 1, \dots, 5\} \cup \{p_{3,i} \mid i = 1, \dots, 9\},$$

the alphabet Σ shall be $\Sigma = \{a, b, c, d\}$, the transition function δ shall be defined as follows:

$$\begin{array}{rll}
\delta(q_0,\varepsilon,0) = (q_1,+1), & \delta(q_1,a,1) = (p_{1,1},+1), & \\
\delta(q_1,b,1) = (p_{2,1},+1), & \delta(q_1,c,1) = (p_{3,1},-1), & \\
\delta(q_1,d,1) = (q_2,-1), & \delta(q_i,\varepsilon,1) = (q_{i+1},-1), & i = 2,\dots,20,\\
\delta(q_{21},\varepsilon,0) = (q_0,0), & \delta(p_{1,i},\varepsilon,1) = (p_{1,i+1},+1), & i = 1,\dots,3,\\
\delta(p_{1,4},\varepsilon,1) = (q_1,+1), & \delta(p_{2,i},\varepsilon,1) = (p_{2,i+1},+1), & i = 1,\dots,4,\\
\delta(p_{2,5},\varepsilon,1) = (q_1,+1), & \delta(p_{3,i},\varepsilon,1) = (p_{3,i+1},+1), & i = 1,\dots,8,\\
\delta(p_{3,9},\varepsilon,1) = (q_1,+1), & &
\end{array}$$

and the set F of accepting states shall consist only of the initial state, i.e., $F = \{q_0\}$. It is not hard to see that this automaton indeed accepts the language L: to arrive from the state q_1 to the only accepting state q_0, it is first obviously necessary to reach a configuration with the state q_1 and with the counter value 20. When the state q_1 is reached for the first time after the last visit of the state q_0, the counter value is always 1. That is, before leaving the state q_1 for the last time before the next visit of q_0, the counter value has to be increased by the value of 19.

There are three "cycles" from the state q_1: one consists of the states $p_{1,i}, i = 1, \ldots, 4$, and the character a is read during the cycle, second of the states $p_{2,i}, i = 1, \ldots, 5$, and the character b is read during the cycle, and third of the states $p_{3,i}, i = 1, \ldots, 9$, and the character c is read. The counter value is increased by the value of 5 in the first cycle, by the value of 6 in the second cycle, and by the value of 8 in the third cycle. The reader may easily convince himself that the only way how to express the number 19 as an integer conical combination of numbers $5, 6$ and 8 is

$$5 + 6 + 8 = 19.$$

Thus, to increase the counter value by the value of 19, the computation has to go through each of these cycles exactly once. However, the third cycle cannot be used as the first of these cycles, since it can be easily seen that if the initial counter value is 1, the computation gets stuck in this cycle. After going through these cycles, the character d is always read. Thus, the statement $L(A) = L$ is proved. Moreover, the above reasoning also clearly implies that the automaton A is strictly transition-$\mathcal{A}$-equiloaded. That is, L is in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$.

Now we shall prove that the language L is not in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C})$. For the purpose of contradiction, let us suppose that a strictly transition-$\mathcal{C}$-equiloaded deterministic one-counter automaton $A' = (K', \Sigma, \delta', q'_0, F')$ exists, such that $L(A') = L$ (we can clearly assume that the alphabet of A' is Σ, since a transition on a character not in Σ would either spoil the strict transition-$\mathcal{C}$-equiloadedness, or change the accepted language).

The requirement of determinism of the automaton A' implies that every computation of the automaton A' begins with a sequence of ε-transitions and then arrives to some configuration with a state q and a counter value t, where it can choose (at least) between some a-transition and some b-transition (but there is not any ε-transition). Moreover, Lemma 3.4.1 implies that $t \geq 1$.

That is, the setting is as follows: for some q in K' and $t \geq 1$, we have $(q_0, \varepsilon, 0) \vdash^+ (q, \varepsilon, t)$. Moreover, for some q_a, q_b in K' and r_a, r_b in $\{-1, 0, +1\}$, $e_a = (q, a, 1, q_a, r_a)$ and $e_b = (q, b, 1, q_b, r_b)$ are transitions of A'.

Since the automaton A' is strictly transition-$\mathcal{C}$-equiloaded, the properties (i) and (ii) of the characterization given in Theorem 3.3.1 have to hold. Since, in addition, an arbitrarily long word beginning by a (resp. b) can be found in $L(A')$, this implies that if the transition e_a is chosen, the computation path has to be able to return to the transition e_b, and vice versa. That is, words w_a and w_b in Σ^* exist, such that

$$(q, aw_a, t) \vdash (q_a, w_a, t + r_a) \vdash^* (q, \varepsilon, t_a) \tag{3.11}$$

and

$$(q, bw_b, t) \vdash (q_b, w_b, t + r_b) \vdash^* (q, \varepsilon, t_b) \tag{3.12}$$

for some t_a, t_b in $\mathbb{N}$. Clearly, the transitions e_a and e_b can be used in an arbitrary order. Thus, it follows from Lemma 3.4.2 that the counter value is always positive in both the computation (3.11) and the computation (3.12). Furthermore, the inequalities $0 < t_a < t$ and $0 < t_b < t$ have to hold, since otherwise either the counter value 0 would be reached, or the "cycles" (3.11) and (3.12) could be "executed" arbitrarily many times and the automaton A' would not be strictly transition-$\mathcal{C}$-equiloaded (the property (ii) of the characterization from Theorem 3.3.1 would be violated).

Now, let us examine the possible lengths of the word w_a. The length of the word w_a clearly cannot be $4k + 3$ for some k in $\mathbb{N}$, since otherwise the computation clearly could continue by the use of some a-transition. However, the only a-transition leading from q at the counter value greater than 0 is e_a. But if this transition was used, the property (i) of the characterization from Theorem 3.3.1 would be violated.

If the length of the word w_a is $4k + 2$ for some k in $\mathbb{N}$, then the "accepting branch" of the computation would have to continue by some d-transition leading from q at the counter value greater than 0. However, the transition e_b would still be unused. This implies that this computation as well as the word w_a can be prolonged so that the prolonged length of w_a is $4k + 1$ or $4k$ for some k in $\mathbb{N}$.

If the length of the word w_a is $4k$ for some k in $\mathbb{N}$, then, clearly, the computation can continue by the use of some c-transition. That is, there is a transition $e_c = (q, c, 1, q_c, r_c)$, for some q_c in K' and r_c in $\{-1, 0, 1\}$. For the same reasons as above, a word w_c in Σ^* has to exist, such that

$$(q, cw_c, t_a) \vdash (q_c, w_c, t_a + r_c) \vdash^* (q, \varepsilon, t_c), \tag{3.13}$$

and such that the counter value is always positive during this computation. However, since $t > t_a$, it is also possible to reverse the order of the cycles, that is, to first use the c-cycle and only then to use the a-cycle. Instead of the word aw_acw_c, the word cw_caw_a would be read, but the terminal configuration would be exactly the same. But since the word aw_acw_c can be prolonged[5] to some word from $L(A')$, the same is true for cw_caw_a. But this is a contradiction, since the first character of this word is c and there is not any such word in $L(A')$.

Thus, the only possible length of the word w_a that is left, is $4k+1$ for some k in $\mathbb{N}$. However, it is clearly possible to construct an arbitrarily long word w in $L(A')$, such that for every k in $\mathbb{N}$, the character $w[4k+2]$, if defined, is c and not b. That is, the number of steps needed until the transition e_b is used in some accepting computation, is not bounded by any constant. This clearly contradicts the characterization given in Theorem 3.3.1.

Now, let us consider the language $L' = \{abcd, acbd, bacd, bcad\}^*\{\varepsilon, c\}$. Clearly, $N(A) = L'$. Furthermore, the automaton A is obviously strictly transition-$\mathcal{E}$-equiloaded. Thus, L' is in $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E})$.

To prove that L' is not in $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$, exactly the same reasoning as above can be used. The only minor difference is in the case when the word w_a is supposed to have the length $4k$ for some k in $\mathbb{N}$. Then, the contradiction is reached not only by noting that a word beginning by c would have been in $L(A')$, but by noting that a word *of length at least 2* beginning by c would have been in $L(A')$ (that is clearly not the case for the language L'). The theorem is proved. $\square$

In what follows, we shall examine the relations of the families of strictly transition-$\mathcal{S}$-equiloaded DOCA-languages to some families of languages studied in the previous chapter. We shall observe that the computational power of (state-accepting) strictly transition-$\mathcal{S}$-equiloaded DOCA is greater than the computational power of strictly transition-$\mathcal{S}$-equiloaded DFA or DFAε. However, on the other hand, we shall also make one slightly less optimistic observation: strictly transition-$\mathcal{S}$-equiloaded DOCA accept only some proper subset of the family of regular languages.

Theorem 3.5.3 The following strict inclusions hold:

1. $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C}) \supsetneq \mathscr{L}_{\delta-SEQ-DFA\varepsilon}$,
2. $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A}) \supsetneq \mathscr{L}_{\delta-SEQ-DFA\varepsilon}$.

Proof. It clearly suffices to prove the first statement: the second statement is a clear corollary of the first statement and of Theorem 3.5.2.

First, let us prove the improper inclusion. Let L in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ be a strictly transition-equiloaded DFAε-language. Recall that $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ is defined by

$$\mathscr{L}_{\delta-SEQ-DFA\varepsilon} := \mathscr{L}_{\delta-SEQ-DFA\varepsilon}(\mathcal{C}) = \mathscr{L}_{\delta-SEQ-DFA\varepsilon}(\mathcal{A}).$$

Thus, there exists a strictly transition-$\mathcal{C}$-equiloaded deterministic finite automaton with ε-transitions $A = (K, \Sigma, \delta, q_0, F)$, such that $L(A) = L$. Moreover, we can clearly suppose that the graphical representation of the automaton A is connected. Thus, by Theorem 2.2.2, the graphical representation of the automaton A either does not contain any reachable directed cycle, or is a directed cycle through all states.

Let us define a strictly transition-$\mathcal{C}$-equiloaded deterministic one-counter automaton $A' = (K', \Sigma', \delta', q_0', F')$ accepting L as follows: $K' = K$, $\Sigma' = \Sigma$, $q_0' = q_0$, $F' = F$, and

$$\forall p, q \in K' \; \forall c \in \Sigma' \cup \{\varepsilon\} : \delta'(p, c, 0) = (q, 0) \iff \delta(p, c) = q$$

[5]This obviously holds for at least one possible w_a, since otherwise the contradiction with $L(A') = L$ could be easily reached. Without loss of generality, let us suppose that we have chosen such w_a.

(transitions for the counter values greater than zero are left undefined). That is, the DOCA A' merely simulates the DFAε A with the counter value being constantly 0. The statement $L(A') = L$ is obvious.

It remains to prove that the deterministic one-counter automaton A' is strictly transition-$\mathcal{C}$-equiloaded. As we have already noted, the graphical representation of the automaton A either does not contain any reachable directed cycle, or is a directed cycle through all states.

If the graphical representation of the automaton A does not contain any reachable directed cycle, then, clearly, there is not any word w in Σ^*, such that $(q_0, w) \vdash_A^* (q_0, \varepsilon)$. By the definition of the automaton A', this implies that there is not any word w in $(\Sigma')^* = \Sigma^*$, such that $(q_0, w, 0) \vdash_{A'}^* (q_0, \varepsilon, 0)$. Thus, the property (i) of the characterization given in Theorem 3.3.1 is trivially satisfied. Moreover, there clearly is a nonnegative integer k in $\mathbb{N}$, such that for every computation path γ of the automaton A, the inequality $|\gamma| \leq k$ holds. Thus, the same property holds also for every computation path of the automaton A'. In other words, the property (ii) of the characterization given in Theorem 3.3.1 is satisfied as well and the deterministic one-counter automaton A' is strictly transition-$\mathcal{C}$-equiloaded.

Similarly, if the graphical representation of the automaton A is a directed cycle through all states, it follows directly from the definition of the automaton A' that the properties (i) and (ii) of the characterization given in Theorem 3.3.1 are satisfied. That is, the deterministic one-counter automaton A' is again strictly transition-$\mathcal{C}$-equiloaded.

Let us prove that the inclusion is proper. In Example 3.1.2, we have observed that the language $L = \{ab, ba\}^*$ is in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C})$. We shall prove that L is not in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$.

For the purpose of contradiction, let us suppose that L is in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. Since the language L is infinite, a DFAε $A = (K, \Sigma, \delta, q_0, F)$ with the graphical representation of the form of a directed cycle through all states exists, such that $L(A) = L$. Let q in K be the first (in the direction of the directed cycle) state of the automaton A, such that at least one non-ε transition (q, c, q') in D leads from q. Such a state has to exist, since otherwise the accepted language $L(A)$ would be finite. If $c = a$, then every word in $L(A)$ begins by a, and that contradicts the assumption $L(A) = L$. An analogous contradiction can be reached also in the case $c = b$. Clearly, these are the only two possibilities. Thus, L is not in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$. The theorem is proved. □

Corollary 3.5.4 The following strict inclusions hold:

1. $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C}) \supsetneq \mathscr{L}_{\delta-SEQ-DFA}$,
2. $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A}) \supsetneq \mathscr{L}_{\delta-SEQ-DFA}$.

Proof. Follows directly from Theorem 3.5.3 and Theorem 2.2.8. □

Theorem 3.5.5 The following relations hold:

1. The families $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$ and $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ are incomparable.
2. The families $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E})$ and $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ are incomparable.
3. The families $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$ and $\mathscr{L}_{\delta-SEQ-DFA}$ are incomparable.
4. The families $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E})$ and $\mathscr{L}_{\delta-SEQ-DFA}$ are incomparable.

Proof. We shall prove all four statements at once. That is, we shall find an example of a language that is in both families of strictly transition-$\mathcal{S}$-equiloaded DOCA-languages accepted by empty memory, mentioned in the statement of the theorem, but not in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ and $\mathscr{L}_{\delta-SEQ-DFA}$, and vice versa.

Let us consider the language $L' = \{ab, ba\}^*\{\varepsilon, aaa, aab, bb\}$ from Example 3.1.3. There, we have already observed that the language L' is both in the family $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$, and in the family $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E})$. However, L' is neither in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$, nor in $\mathscr{L}_{\delta-SEQ-DFA}$, since it contains two words beginning by different characters.

On the other hand, let us consider the language $L = \{a\}$. The language L is finite, and thus clearly is both in $\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$ and in $\mathscr{L}_{\delta-SEQ-DFA}$. However, obviously, every language accepted by some deterministic one-counter automaton by empty memory has to contain the empty word ε. Thus, L is neither in $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$, nor in $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E})$. The theorem is proved. □

Theorem 3.5.6 The following strict inclusions hold:

1. $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C}) \subsetneq \mathscr{R}$,
2. $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A}) \subsetneq \mathscr{R}$,
3. $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C}) \subsetneq \mathscr{R}$,
4. $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E}) \subsetneq \mathscr{R}$.

Proof. It follows from Theorem 3.5.1 and Theorem 3.5.2 that it suffices to prove the second strict inclusion $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A}) \subsetneq \mathscr{R}$, and the remaining strict inclusions will follow as a direct corollary.

First, let us prove the inclusion. Let L be a language in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$. Then, a strictly transition-$\mathcal{A}$-equiloaded deterministic one-counter automaton $A = (K, \Sigma, \delta, q_0, F)$ exists, such that $L(A) = L$.

Since the automaton A is strictly transition-$\mathcal{A}$-equiloaded, the properties (i) and (ii') of the characterization from Theorem 3.3.1 have to be satisfied. Let us denote by H^* the highest possible nonnegative integer, such that a configuration with the counter value H^* is reachable by A. Clearly, the properties (i) and (ii'), together with Lemma 3.2.1 imply that $H^* \leq |K|(|K|+1) =: M$.

Thus, we may construct a DFAε $A' = (K', \Sigma', \delta', q_0', F')$ accepting the language L as follows: $K' = K \times \{0, \ldots, M\}$, $\Sigma' = \Sigma$, $q_0' = (q_0, 0)$, $F' = F \times \{0, \ldots, M\}$, and for all q, q' in K, all t, t' in $\{0, \ldots, M\}$, and all c in $\Sigma \cup \{\varepsilon\}$:

$$\delta'((q,t),c) = (q',t') \iff \delta(q,c,\operatorname{sgn}(t)) = (q',r),$$

where $t' = t + r$. The transitions that would increase the counter value of the automaton A above the value of M shall remain undefined for A'. We consider the statement $L(A') = L(A) = L$ to be obvious.

Several examples of regular languages that are not strictly transition-$\mathcal{A}$-equiloaded, shall be constructed in the subsection dealing with closure properties. Thus, the above proved inclusion is strict, and the theorem is proved. □

Now, let us turn our attention to the families of strictly state-$\mathcal{S}$-equiloaded DOCA-languages.

Theorem 3.5.7 The following relations hold:

1. The families $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$ and $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$ are incomparable.
2. The families $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$ and $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$ are incomparable.
3. The families $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$ and $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$ are incomparable.
4. The families $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$ and $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$ are incomparable.

Proof. We shall prove all four incomparability relations at once. In Example 3.1.6, we have observed that the (obviously non-prefix-dense) language

$$L_1 = \{a^n b \mid n \geq 1\} \cup \{\varepsilon\}$$

is a member of the family $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$ and thus, as a consequence of Theorem 1.6.3, also of the family $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$. However, since the language L_1 is not prefix-dense, it follows from Theorem 1.7.5 that it is neither in the family $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$, nor in the family $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$.

On the other hand, let us consider the finite language $L_2 = \{a\}$. It is trivial to construct a strictly state-$\mathcal{C}$-equiloaded deterministic one-counter automaton accepting L_2 by accepting state.

Thus, L_2 is in the family $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$ and, as a consequence of Theorem 1.6.3, also in the family $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$. However, since L_2 does not contain the empty word ε, it cannot be accepted by any deterministic one-counter automaton by empty memory, and thus, L_2 is neither the member of the family $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$, nor of the family $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$. The theorem is proved. □

Theorem 3.5.8 The following strict inclusion holds:

$$\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C}) \subsetneq \mathscr{N}_{K-SEQ-DOCA}(\mathcal{E}).$$

Proof. The inclusion follows immediately from Theorem 1.6.3. We shall prove that this inclusion is strict.

Let us consider the language $L = \{a^n b^n \mid n \geq 0\}$. In Example 3.1.4, we have observed that the language L is in $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$. We shall prove that the language L is not in the family $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$.

For the purpose of contradiction, let us suppose that $A = (K, \Sigma, \delta, q_0, F)$ is a strictly state-$\mathcal{C}$-equiloaded deterministic one-counter automaton, such that $N(A) = L$. Since the language L is not regular, for every t in $\mathbb{N}$ a nonnegative integer m in $\mathbb{N}$ exists, such that the counter value t is reached during the computation on a word $a^m b^m$. This, together with the strict state-$\mathcal{C}$-equiloadedness of the automaton A, implies that[6] a nonnegative integer n in $\mathbb{N}$ exists, such that the computation of the automaton A on the word $a^n b^n$ has a form

$$(q_0, a^n b^n, 0) \vdash^+ (q, a^{n_1} b^n, t_1) \vdash^+ (q, a^{n_2} b^n, t_2) \vdash^+ (q, b^{n_3}, t_3) \vdash^+ (q, b^{n_4}, t_4) \vdash^+ (q', \varepsilon, 0),$$

where q and q' in K are states, n_1, n_2, n_3 and n_4 in $\mathbb{N}$ are nonnegative integers, such that $0 < n_2 < n_1 < n$ and $0 < n_4 < n_3 < n$, t_1, t_2, t_3 and t_4 in $\mathbb{N}$ are counter values, such that $0 < t_1 < t_2$, $0 < t_4 < t_3$ and $t_1 \leq t_4$, and the counter value is always positive during the computation $(q, a^{n_1} b^n, t_1) \vdash^* (q, b^{n_4}, t_4)$.

However, this clearly implies that positive integers d_1, d_2 in $\mathbb{N}^+$ exist,[7] such that the word $w = a^n b^{n-n_4} a^{(n_1-n_2)d_1} b^{(n_3-n_4)d_2} b^{n_4}$ is in $N(A)$. However, clearly, w is not in L. Thus, $N(A) \neq L$, and that is a contradiction with our assumption. The theorem is proved. □

We leave the relation between the analogous families of languages accepted by accepting state open.

Open Problem 3.5.9 What is the relation between the family $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$ and the family $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$? Clearly, Theorem 1.6.3 implies $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C}) \subseteq \mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$. Is this inclusion proper or are these two families equal?

Theorem 3.5.10 The following strict inclusions hold:

1. $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C}) \supsetneq \mathscr{L}_{K-SEQ-DFA\varepsilon}$,
2. $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A}) \supsetneq \mathscr{L}_{K-SEQ-DFA\varepsilon}$.

Proof. We shall prove both strict inclusions at once. First, let us prove the inclusions. Every strictly state-equiloaded deterministic finite automaton with ε-transitions can be simulated by a strictly state-$\mathcal{C}$-equiloaded (and thus also strictly state-$\mathcal{A}$-equiloaded) deterministic one-counter automaton with the counter value being constantly zero. Details of this construction are as in the proof of Theorem 3.5.3.

Now, let us prove that these inclusions are strict. In Example 3.1.5, we have observed that both the family $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$ and the family $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$ contains some nonregular language. However, a nonregular language cannot be in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ neither. The theorem is proved. □

Corollary 3.5.11 The following strict inclusions hold:

[6] The proof of the following statement is straightforward and since similar ideas have already been used a number of times in this thesis, we consider it to be clear. The details are therefore left to be worked out by the reader (if needed).

[7] These positive integers can be defined to be positive integers d_1, d_2, such that $(t_2 - t_1)d_1 = (t_3 - t_4)d_2$.

1. $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C}) \supsetneq \mathscr{L}_{K-SEQ-DFA}$,
2. $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A}) \supsetneq \mathscr{L}_{K-SEQ-DFA}$.

Proof. The statement is a direct corollary of Theorem 3.5.10 and of Theorem 2.2.8. □

Theorem 3.5.12 The following relations hold:

1. The families $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$ and $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ are incomparable.
2. The families $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$ and $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ are incomparable.
3. The families $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$ and $\mathscr{L}_{K-SEQ-DFA}$ are incomparable.
4. The families $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$ and $\mathscr{L}_{K-SEQ-DFA}$ are incomparable.

Proof. We shall prove all four incomparability relations at once. In Example 3.1.6, we have observed that the non-prefix-dense language

$$L = \{a^n b \mid n \geq 1\} \cup \{\varepsilon\}$$

is both in $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$, and in $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$. However, it is a direct consequence of Theorem 1.7.5 that the language L is neither in $\mathscr{L}_{K-SEQ-DFA\varepsilon}$, nor in $\mathscr{L}_{K-SEQ-DFA}$.

On the other hand, clearly, the language $L' = \{a\}$ is both in the family $\mathscr{L}_{K-SEQ-DFA\varepsilon}$ and in the family $\mathscr{L}_{K-SEQ-DFA}$. However, since it does not contain the empty word ε, it cannot be neither in $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$, nor in $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$. The theorem is proved. □

Theorem 3.5.13 The following relations hold:

1. The families $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$ and $\mathscr{R}$ are incomparable.
2. The families $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$ and $\mathscr{R}$ are incomparable.
3. The families $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$ and $\mathscr{R}$ are incomparable.
4. The families $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$ and $\mathscr{R}$ are incomparable.

Proof. In all four cases, let us disprove both possible inclusions. In Example 3.1.5, we have observed that the nonregular language

$$L_1 = \{\, w \in \{a,b\}^* \mid \forall u \in \{a,b\}^*, u \text{ is prefix of } w : \#_a(u) \geq \#_b(u)\}$$

is both in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$, and in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$. In Example 3.1.4, we have observed that the nonregular language

$$L_2 = \{\, a^n b^n \mid n \geq 0\}$$

is in $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$. Finally, let us consider the state-$\mathcal{C}$-equiloaded deterministic one-counter automaton A from example 3.1.5, with the property $L(A) = L_1$. Clearly,

$$N(A) = \{\, w \in \{a,b\}^* \mid \#_a(w) = \#_b(w) \text{ and } \forall u \in \{a,b\}^*, u \text{ is prefix of } w : \#_a(u) \geq \#_b(u)\}\,.$$

This language is obviously nonregular, however is in $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$.

To disprove the converse inclusions, let us consider an arbitrary regular non-prefix-dense language, e.g., $L_3 = \{a\}^*\{b\}$. By Theorem 1.7.5, this language cannot be neither in the family $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$, nor in the family $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$. Further, the language L_3 does not contain the empty word ε.[8] Thus, it is not neither in $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$, nor in $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$. The theorem is proved. □

[8]In contrast to our previous statement, this is not the general property of regular non-prefix-dense languages. In fact, in Example 3.1.6, we have observed that there is a language that is non-prefix-dense, regular, contains ε, and is both in $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$ and in $\mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$.

In what follows, we shall focus on the relation between the families of strictly state-$\mathcal{S}$-equiloaded DOCA-languages and the families of strictly transition-$\mathcal{S}$-equiloaded DOCA-languages. In Theorem 2.2.9, we have observed that in the case of deterministic finite automata, strictly state-equiloaded automata have greater computational power than strictly transition-equiloaded automata. In the following theorem, we shall prove that this observation generalizes also to the case of deterministic one-counter automata.

Theorem 3.5.14 The following strict inclusions hold:

1. $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C}) \subsetneq \mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$,
2. $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A}) \subsetneq \mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$,
3. $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C}) \subsetneq \mathscr{N}_{K-SEQ-DOCA}(\mathcal{C})$,
4. $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E}) \subsetneq \mathscr{N}_{K-SEQ-DOCA}(\mathcal{E})$.

Proof. We shall prove all four statements at once. First, let us prove the improper inclusions. Let $A = (K, \Sigma, \delta, q_0, F)$ be a strictly transition-$\mathcal{S}$-equiloaded deterministic one-counter automaton, for some $\mathcal{S}$ in $\{\mathcal{C}, \mathcal{A}, \mathcal{E}\}$. We shall construct a strictly state-$\mathcal{S}$-equiloaded deterministic one-counter automaton $A' = (K', \Sigma', \delta', q_0', F')$, such that $L(A') = L(A)$, and $N(A') = N(A)$.

The idea behind the construction shall be as follows: the set of states of the automaton A' shall be precisely the set of transitions of the automaton A. Moreover, the following invariant shall be hold: if a nonempty computation path γ of the automaton A on a word w ends by going through a transition e in D_A and with the counter value being t, then the corresponding computation path γ' of the automaton A' on the word w ends in the *state* e in $K' = D_A$ and with the couter value being t as well.

Two further details need to be addressed. The initial state of the automaton A' shall be an arbitrary transition of the automaton A leading to the state q_0 (we shall suppose that such a transition exists; otherwise, the accepted language would be necessarily finite and a discussion would become more-or-less trivial). Finally, the set of accepting states of the automaton A' shall be the set of all transitions of the automaton A leading to accepting states.

The formal construction is as follows: $K' = D_A$, $\Sigma' = \Sigma$, the transition function δ' is defined for p, q in K, c, d in Σ, t, t' in $\{0, 1\}$ and r' in $\{-1, 0, 1\}$ by

$$\delta'((p, c, t', q, r'), d, t) = ((q, d, t, \mathrm{pr}_1(\delta(q, d, t)), \mathrm{pr}_2(\delta(q, d, t))), \mathrm{pr}_2(\delta(q, d, t)))$$

if $\delta(q, d, t)$ is defined for the automaton A. If it is not defined, then the transition function δ' is left undefined for the corresponding inputs as well. Moreover, q_0' is defined to be an arbitrary member of the set

$$\{(p, c, t, q, r) \in D_A \mid q = q_0\}.$$

Finally, the set of accepting states F' is defined by

$$F' = \{(p, c, t, q, r) \in D_A \mid q \in F\}.$$

It is obvious that the above mentioned invariant holds. Thus, it is not hard to see that indeed $L(A') = L(A)$ and $N(A') = N(A)$, and that the automaton A' is strictly state-$\mathcal{S}$-equiloaded (assuming that A is strictly transition-$\mathcal{S}$-equiloaded).

Now, let us prove that these inclusions are proper. The statement is a direct corollary of Theorem 3.5.6 and of Theorem 3.5.13. To be more specific, in Theorem 3.5.13 we have proved that each of the four families on the right side of our relations contain at least one nonregular language. However, in Theorem 3.5.6 we have proved that none of the four families of languages on the left side contain any nonregular language. The theorem is proved. □

3.6 Closure Properties

In this section, we shall prove some of the closure properties of the families of strictly $\mathcal{S}$-equiloaded DOCA-languages. The closure properties that are not stated in this section are open up to now.

Closure Properties of the Families of Strictly Transition-$\mathcal{S}$-Equiloaded DOCA-Languages

Now, we shall concentrate on closure properties of the families of strictly transition-$\mathcal{S}$-equiloaded languages, $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C})$, $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$, $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$, and, finally, the family $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E})$. By Theorem 3.5.1 and Theorem 3.5.2, to prove that none of these families is closed under a certain operation, it suffices to find operands,[9] such that they are in $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$, but the result of the operation applied to these operands is not a member of the family $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$. This is the case because the fact that a language is in the family $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$ implies that it is also in the other three families. Conversely, if a language is not in the family $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$, it also cannot be in any of the remaining three families.

Theorem 3.6.1 None of the families $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C})$, $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$, $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$, and $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E})$ is closed under concatenation.

Proof. Let us consider the languages $L_1 = \{a\}^*$, and $L_2 = \{\varepsilon, b\}$. Clearly, both the language L_1 and the language L_2 are in $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$: the construction of strictly transition-$\mathcal{C}$-equiloaded deterministic one-counter automata accepting these languages by empty memory is left to the reader as a more-or-less trivial exercise.

We shall prove that the concatenation of these languages, the language

$$L_1 \cdot L_2 = \{a^n b^i \mid n \in \mathbb{N}, i \in \{0,1\}\},$$

is not in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$. For the purpose of contradiction, let us suppose that the language $L_1 \cdot L_2$ is in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$. Then, a strictly transition-$\mathcal{A}$-equiloaded deterministic one-counter automaton $A = (K, \Sigma, \delta, q_0, F)$ exists, such that $L(A) = L_1 \cdot L_2$. Since A is strictly transition-$\mathcal{A}$-equiloaded, a nonnegative integer constant k in $\mathbb{N}$ exists, such that the inequality

$$|\#[e,\gamma] - \#[f,\gamma]| \leq k \tag{3.14}$$

holds for all transitions e, f in D and all accepting computation paths γ of the automaton A. Let us denote the number of a-transitions of the automaton A by m. Let us consider some accepting computation path γ_w corresponding to some accepting computation on the word $w = a^{km+1}$. It follows from the Pigeonhole principle that an a-transition e_a in D exists, such that

$$\#[e_a, \gamma_w] \geq k+1.$$

Furthermore, since there is a word containing the character b in the accepted language $L(A)$, the automaton A has to have at least one b-transition e_b. However, clearly,

$$\#[e_b, \gamma_w] = 0.$$

Thus,

$$|\#[e_a, \gamma_w] - \#[e_b, \gamma_w]| \geq k+1. \tag{3.15}$$

However, (3.15) contradicts (3.14), and thus, the language $L_1 \cdot L_2$ is not a member of the family $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$. □

Theorem 3.6.2 None of the families $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C})$, $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$, $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$, and $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E})$ is closed under union.

Proof. Let us consider the same counterexample as in the proof of the previous theorem, i.e., the languages $L_1 = \{a\}^*$ and $L_2 = \{\varepsilon, b\}$.

By a very similar reasoning as in the proof of the previous theorem, we shall prove that the language $L_1 \cup L_2$ is not in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$. For the purpose of contradiction, let us suppose that the language $L_1 \cup L_2$ is in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$. Then, a strictly transition-$\mathcal{A}$-equiloaded deterministic one-counter automaton $A = (K, \Sigma, \delta, q_0, F)$ exists, such that $L(A) = L_1 \cup L_2$. Since

[9] Or *an operand*, in the case of unary operations.

A is strictly transition-$\mathcal{A}$-equiloaded, a nonnegative integer constant k in $\mathbb{N}$ exists, such that the inequality

$$|\#[e,\gamma] - \#[f,\gamma]| \leq k \tag{3.16}$$

holds for all transitions e, f in D and for all accepting computation paths γ of the automaton A. Let m denote the number of a-transitions of the automaton A, and let us consider some accepting computation path γ_w corresponding to some accepting computation on the word $w = a^{km+1}$. By the Pigeonhole principle, an a-transition e_a in D exists, such that

$$\#[e_a, \gamma_w] \geq k + 1.$$

As in the proof of the previous theorem, the automaton A has to have at least one b-transition e_b, such that

$$\#[e_b, \gamma_w] = 0.$$

Thus,

$$|\#[e_a, \gamma_w] - \#[e_b, \gamma_w]| \geq k + 1. \tag{3.17}$$

However, the inequality (3.17) contradicts the inequality (3.16). □

Theorem 3.6.3 None of the families $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C})$, $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$, $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$, and $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E})$ is closed under complementation.

Proof. Let us consider the language $L = \{\varepsilon, a\}$ over the alphabet $\Sigma = \{a\}$. Clearly, the language L is a member of the family of languages $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$.

We shall prove that the complement of the language L, the language

$$L^C = \{a^k \mid k \geq 2\}$$

is not in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$. For the purpose of contradiction, let us suppose that the language L^C is in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$. Then, a strictly transition-$\mathcal{A}$-equiloaded deterministic one-counter automaton $A = (K, \Sigma, \delta, q_0, F)$ exists, such that $L(A) = L^C$. For every i in $\mathbb{N}$, let us define the set of states

$$K^i = \{q \in K \mid \exists t \in \mathbb{N} : (q_0, a^i, 0) \vdash^* (q, \varepsilon, t)\}.$$

Clearly, none of the states in K^0 and K^1 is accepting.

Since the language L^C is infinite, it follows from the characterization given in Theorem 3.3.1 that a nonnegative integer $j \geq 1$ in $\mathbb{N}$ exists, such that

$$(q_0, a^j, 0) \vdash^* (q_0, \varepsilon, 0).$$

Then, since the automaton A is deterministic, we have

$$K^{j+1} = K^1.$$

Thus, none of the states in K^{j+1} is accepting. However, by the definition of K^{j+1}, this implies that a^{j+1} is not in $L(A)$. However, since $j \geq 1$, this contradicts our assumption that $L(A) = L^C$. □

Theorem 3.6.4 None of the families $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C})$, $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$, $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$, and $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E})$ is closed under closure.

Proof. Let us consider the language $L = \{\varepsilon, a, b\}$. Obviously, the language L is in the family of languages $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$.

We shall prove that the language $L^* = \{a, b\}^*$ is not in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$. For the purpose of contradiction, let us suppose that L^* is in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$. Then, a strictly transition-$\mathcal{A}$-equiloaded deterministic one-counter automaton $A = (K, \Sigma, \delta, q_0, F)$ exists, such that $L(A) = L^*$.

Since the automaton A is strictly transition-$\mathcal{A}$-equiloaded, a nonnegative integer constant k in $\mathbb{N}$ exists, such that

$$|\#[e,\gamma] - \#[f,\gamma]| \leq k \tag{3.18}$$

holds for all transitions e, f in D and all accepting computation paths γ. Let us denote by m the number of a-transitions in D, and let us consider some accepting computation path γ_w corresponding to some accepting computation on the word $w = a^{km+1}$. By the Pigeonhole principle, an a-transition e_a in D exists, such that

$$\#[e_a, \gamma_w] \geq k+1.$$

Furthermore, since $L(A) = L^*$, at least one b-transition e_b has to exist in D. Clearly,

$$\#[e_b, \gamma_w] = 0.$$

Thus,

$$|\#[e_a, \gamma_w] - \#[e_b, \gamma_w]| \geq k+1. \tag{3.19}$$

The inequality (3.19) contradicts (3.18). □

Theorem 3.6.5 None of the families $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C})$, $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$, $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$, and $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E})$ is closed under positive closure.

Proof. Exactly the same counterexample, i.e., the language $L = \{\varepsilon, a, b\}$, and exactly the same reasoning can be used as in the proof of Theorem 3.6.4. □

Theorem 3.6.6 None of the families $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C})$, $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$, $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$, and $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E})$ is closed under reversal.

Proof. Let us consider the language $L = \{ab\}^*\{\varepsilon, a\}$. It is not hard to construct a strictly transition-$\mathcal{C}$-equiloaded deterministic one-counter automaton accepting L by empty memory. That is, L is in $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$.

We shall prove that the language $L^R = \{\varepsilon, a\}\{ba\}^*$ is not in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$. For the purpose of contradiction, let us suppose that L^R is in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$. Then, a strictly transition-$\mathcal{A}$-equiloaded deterministic one-counter automaton $A = (K, \Sigma, \delta, q_0, F)$ exists, such that $L(A) = L^R$. The properties (i) and (ii') of the characterization given in Theorem 3.3.1 have to be satisfied by the automaton A.

By the infiniteness of the language $L(A) = L^R$, and by the property (ii') of this characterization, at least one word w in Σ^+ exists, such that $(q_0, w, 0) \vdash^+ (q_0, \varepsilon, 0)$. If every such word was of the form ax for some x in Σ^*, then there would be only a finite number of words in $L(A)$ beginning by b, and that would contradict our assumption $L(A) = L^R$. An analogous contradiction can be reached also in the case when every such word is of the form by for some y in Σ^*.

Thus, a word u in Σ^* exists, such that $(q_0, au, 0) \vdash^+ (q_0, \varepsilon, 0)$, and a word v in Σ^* exists, such that $(q_0, bv, 0) \vdash^+ (q_0, \varepsilon, 0)$.

Let the word au end with the character a. Then, the word $auau$ contains at least one occurence of the subword aa. Furthermore,

$$(q_0, auau, 0) \vdash^+ (q_0, au, 0) \vdash^+ (q_0, \varepsilon, 0).$$

Since the language $L(A)$ is infinite, it is possible to reach an accepting configuration from some configuration with the state q_0 and the counter value 0. In other words, a word z in Σ^* exists, such that $(q_0, z, 0) \vdash^* (q, \varepsilon, t)$, where q in F is an accepting state and t in $\mathbb{N}$ is a nonnegative integer. This implies that

$$(q_0, auauz, 0) \vdash^+ (q_0, auz, 0) \vdash^+ (q_0, z, 0) \vdash^* (q, \varepsilon, t),$$

i.e., $auauz$ is in $L(A)$. However, since aa is a subword of $auauz$, this contradicts our assumption that $L(A) = L^R$.

Thus, the word au ends with the character b. Then, the word $aubv$ contains at least one occurence of the subword bb. Similarly as in the previous case, it is possible to prove that

$$(q_0, aubvz, 0) \vdash^+ (q_0, bvz, 0) \vdash^+ (q_0, z, 0) \vdash^* (q, \varepsilon, t),$$

where z is in Σ^*, q is in F, and t is in $\mathbb{N}$. Consequently, $aubvz$ is in $L(A)$, and that is again a contradiction with our assumption that $L(A) = L^R$. □

Theorem 3.6.7 None of the families $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C})$, $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$, $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$, and $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E})$ is closed under inverse homomorphism.

Proof. Let us consider the language $L = \{\varepsilon, a\}$, and the homomorphism $h : \{a,b\}^* \to \{a,b\}^*$ defined as follows:

$$h(a) = a,$$
$$h(b) = \varepsilon.$$

The language L is clearly in $\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$. We shall prove that the language

$$h^{-1}(L) = \{w \in \{a,b\}^* \mid \#_a(w) \leq 1\}$$

is not in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$. For the purpose of contradiction, let us suppose that $h^{-1}(L)$ is in $\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$. Then, a strictly transition-$\mathcal{A}$-equiloaded deterministic one-counter automaton $A = (K, \Sigma, \delta, q_0, F)$ exists, such that $L(A) = h^{-1}(L)$. For the automaton A, a nonnegative integer constant k in $\mathbb{N}$ exists, such that for every two transitions e, f in D, and for all accepting computation paths γ, the inequality

$$|\#[e,\gamma] - \#[f,\gamma]| \leq k \tag{3.20}$$

has to hold. Let us denote the number of b-transitions of the automaton A by m, and let us consider some accepting computation path γ_w corresponding to some accepting computation on the word $w = b^{mk+1}$. By the Pigeonhole principle, a b-transition e_b in D exists, such that

$$\#[e_b, \gamma_w] \geq k + 1.$$

Furthermore, $L(A)$ contains a word u such that $\#_a(u) > 0$. Thus, the automaton A has to have at least one a-transition e_a. However, clearly,

$$\#[e_a, \gamma_w] = 0.$$

Thus, we have

$$|\#[e_b, \gamma_w] - \#[e_a, \gamma_w]| \geq k + 1, \tag{3.21}$$

and that is a clear contradiction with (3.20). □

Closure Properties of the Families of Strictly State-$\mathcal{S}$-Equiloaded DOCA-Languages

In what follows, we shall discuss some of the closure properties of the families of strictly state-$\mathcal{S}$-equiloaded DOCA-languages.

Theorem 3.6.8 The families $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$ and $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$ are not closed under concatenation.

Proof. Let us consider the languages $L_1 = \{a\}^*$ and $L_2 = \{b\}$. It is a trivial task to construct strictly state-$\mathcal{C}$-equiloaded deterministic one-counter automata accepting these languages. Thus, L_1 and L_2 are both in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$, and in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$. However, the concatenation $L_1 \cdot L_2$ of these languages is clearly not prefix-dense, and thus, by Theorem 1.7.5, is neither in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$, nor in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$. □

Theorem 3.6.9 The families $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$ and $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$ are not closed under complementation.

Proof. Let us consider the language $L = \{a^n b \mid n \geq 0\}^C$. We shall show that this language is both in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$ and in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$. Let us define a deterministic one-counter automaton $A = (K, \Sigma, \delta, q_0, F)$ as follows: $K = \{q_0, q_1\}$, $\Sigma = \{a, b\}$,

$$\begin{array}{lll} \delta(q_0, a, 0) = (q_1, 0), & \delta(q_0, b, 0) = (q_1, +1), & \delta(q_1, \varepsilon, 0) = (q_0, 0), \\ \delta(q_0, \varepsilon, 1) = (q_1, 0), & \delta(q_1, a, 1) = (q_0, 0), & \delta(q_1, b, 1) = (q_0, 0), \end{array}$$

and $F = \{q_0\}$. Clearly, $L(A) = L$. Moreover, since every computation path of the automaton A alternates between states q_0 and q_1, the automaton A is strictly state-$\mathcal{C}$-equiloaded, and thus also strictly state-$\mathcal{A}$-equiloaded. Thus, L is both $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$ and in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$.

However, the language $L^C = \{a^n b \mid n \geq 0\}$ is not prefix-dense, and thus, by Theorem 1.7.5, is neither in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$, nor in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$. $\square$

Theorem 3.6.10 The families $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$ and $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$ are not closed under reversal.

Proof. Let us consider the language $L = \{ba^n \mid n \geq 0\}$. This language is both in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$ and in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$, since it can be accepted by the automaton $A = (K, \Sigma, \delta, q_0, F)$ defined as follows: $K = \{q_0, q_1\}$, $\Sigma = \{a, b\}$,

$$\delta(q_0, b, 0) = (q_1, +1), \qquad \delta(q_1, a, 1) = (q_0, 0), \qquad \delta(q_0, \varepsilon, 1) = (q_1, 0),$$

and $F = \{q_1\}$. This automaton is clearly strictly state-$\mathcal{C}$-equiloaded, and thus also strictly state-$\mathcal{A}$-equiloaded.

However, the reversal of the language L, the language $L^R = \{a^n b \mid n \geq 0\}$, clearly is not prefix-dense and thus, by Theorem 1.7.5, is neither in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$, nor in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$. $\square$

Theorem 3.6.11 The families $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$ and $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$ are not closed under inverse homomorphism.

Proof. The finite language $L = \{b\}$ is clearly both in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$ and in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$. Now, let us consider the homomorphism $h : \{a, b\}^* \to \{a, b\}^*$ defined by

$$\begin{aligned} h(a) &= \varepsilon, \\ h(b) &= b. \end{aligned}$$

It is clear that the language $h^{-1}(L) = \{w \in \{a, b\}^* \mid \#_b(w) = 1\}$ is not prefix-dense, and thus, by Theorem 1.7.5, is neither in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$, nor in $\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$. $\square$

Conclusion

In this report, we have presented and analyzed several definitions of balanced use of resources in deterministic sequential computations. The main aims of this report have been to generalize the earlier definitions for DFA, presented in [26], [27] and [25] to higher models of computation, to unify the earlier theories of state-equiloadedness and transition-equiloadedness for DFA, and to initiate the study of equiloadedness for some model of computation accepting nonregular sets. We may conclude that these aims have been successfully achieved.

In Section 1.2, we have defined a new abstract model of computation, the *abstract deterministic automata* (ADA). ADA are an AFA-inspired abstraction of deterministic automata with a one-way input tape and some kind of auxiliary memory. Many widely used models of computation, e.g., DFA, DFAε, DOCA, DPDA, or some variants of deterministic Turing machines, can be viewed as special cases of ADA. This has enabled us to present our definitions of equiloadedness independently from a particular model of computation – the definition for ADA applies to all models of computation that we examine in this report.

Next, in Section 1.4 and Section 1.5, we have presented our definitions of *strict $\mathcal{S}$-equiloadedness*, *$\mathcal{S}$-equiloadedness*, and *weak $\mathcal{S}$-equiloadedness*. All of these definitions have been presented for ADA and both for states and for transitions. These definitions generalize the earlier definitions from [25], [26] and [27]. This generalization is twofold: first, the definitions are stated for ADA instead of DFA. Second, in the earlier works, only accepting computation paths have been taken into account. Our definitions include the parameter $\mathcal{S}$ that makes it possible to specify the set of computation paths that we are interested in.

Further, we have examined some properties of equiloadedness that hold for abstract deterministic automata in general. Most importantly, we have proved some relations between the families of equiloaded languages that hold for every model of computation that is a special case of ADA. Moreover, we have defined the concept of prefix-dense languages that can be used to prove that a given language is not strictly $\mathcal{S}$-equiloaded for any model of computation that is a special case of ADA.

Later in this report, we have studied several families of equiloaded DFA, DFAε and DOCA, as well as the corresponding families of equiloaded languages. Equiloaded DFA have been studied already in the earlier works [26], [27] and [25], however we have presented some new results. Equiloaded DFAε and DOCA have not been studied yet.

For several families of equiloaded automata, we have proved their characterizations. Table Concl.1 contains the summary of these characterizations, including the numbers of corresponding theorems. Among characterizations, probably the most important results are the characterization of weakly state-$\mathcal{C}_=$-equiloaded DFA and DFAε (Theorem 2.3.31) and the characterization of strictly transition-$\mathcal{S}$-equiloaded DOCA for $\mathcal{S}$ in $\{\mathcal{C}, \mathcal{A}, \mathcal{E}\}$ (Theorem 3.3.1). Both these characterizations are completely new. The first of these characterization emphasizes the importance of Perron-Frobenius eigenvalues related to DFA and DFAε as a characteristic that can be used to analyze various quantitative properties of finite automata. The second is up to now the only known characterization of equiloaded DOCA. Since it characterizes the families of strictly transition-$\mathcal{S}$-equiloaded DOCA by some properties of computation paths, the decidability of this characterization have not been immediately clear. However, we have proved this decidability and presented an algorithm that decides if a given DOCA is strictly transition-$\mathcal{S}$-equiloaded by deciding if the conditions imposed in the characterization from Theorem 3.3.1 are satisfied.

<table>
<tr><th>The Model</th><th>Type of Equiloadedness</th><th>Resource</th><th>For $\mathcal{S}$ in</th><th>Characterization</th></tr>
<tr><td rowspan="8">DFA and DFAε</td><td rowspan="2">Strict $\mathcal{S}$-Equiloadedness</td><td>States</td><td>$\mathcal{C}, \mathcal{A}$</td><td>Theorem 2.2.1</td></tr>
<tr><td>Transitions</td><td>$\mathcal{C}, \mathcal{A}$</td><td>Theorem 2.2.2</td></tr>
<tr><td rowspan="4">Weak $\mathcal{S}$-Equiloadedness</td><td rowspan="2">States</td><td>$\mathcal{C}_=$</td><td>Theorem 2.3.31</td></tr>
<tr><td>$\mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq$</td><td>?</td></tr>
<tr><td rowspan="2">Transitions</td><td>$\mathcal{C}_=, \mathcal{A}_=$</td><td>Theorem 2.3.18</td></tr>
<tr><td>$\mathcal{C}_\leq, \mathcal{A}_\leq$</td><td>Theorem 2.3.19</td></tr>
<tr><td rowspan="2">$\mathcal{S}$-Equiloadedness</td><td>States</td><td>$\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq$</td><td>?</td></tr>
<tr><td>Transitions</td><td>$\mathcal{C}_=, \mathcal{A}_=, \mathcal{C}_\leq, \mathcal{A}_\leq$</td><td>?</td></tr>
<tr><td rowspan="2">DOCA</td><td rowspan="2">Strict $\mathcal{S}$-Equiloadedness</td><td>States</td><td>$\mathcal{C}, \mathcal{A}, \mathcal{E}$</td><td>?</td></tr>
<tr><td>Transitions</td><td>$\mathcal{C}, \mathcal{A}, \mathcal{E}$</td><td>Theorem 3.3.1</td></tr>
</table>

Table Concl.1: The summary of theorems providing characterizations of the families of equiloaded automata studied in this report.

In Section 2.1, we have observed that several basic quantities used in our study of $\mathcal{S}$-equiloaded DFA and DFAε may be computed as solutions to initial value problems for homogeneous systems of first-order linear OΔEs with constant coefficients. Moreover, we have observed some relations between the matrices of these systems and the transition matrix of a given automaton. Since systems of this kind can be solved relatively easily, this have lead us to the elegant mathematical method of computing closed forms of these basic quantities, and to the numerical algorithm of computing equiloadedness measures for DFA and DFAε. Furthermore, since the matrices of the presented systems are all nonnegative, the results obtained in Section 2.1 have enabled us to use the Perron-Frobenius theory to study the asymptotic properties of these basic quantities.

In Subsection 2.3.1, we have proved that the alternative definition of $\mathcal{S}$-equiloaded DFA and DFAε, based on the definitions given in [26] and [27], is equivalent to our definition of $\mathcal{S}$-equiloadedness. This result unifies the earlier theory of state-equiloaded DFA ([26] and [27]) with the earlier theory of transition-equiloaded DFA ([25]).

$$\begin{tikzcd}
\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C}) \arrow[rr] & & \mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A}) \arrow[r] & \mathscr{R} \\
& \mathscr{L}_{\delta-SEQ-DFA\varepsilon} \arrow[ul] \arrow[ur] & & \\
& \mathscr{L}_{\delta-SEQ-DFA} \arrow[u] & & \\
\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C}) \arrow[uuu] \arrow[rr] \arrow[uur, dash, dotted] \arrow[ur, dash, dotted] & & \mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E}) \arrow[uuu] \arrow[uul, dash, dotted] \arrow[ul, dash, dotted] &
\end{tikzcd}$$

Figure Concl.1: The diagram of relations between various families of strictly transition-$\mathcal{S}$-equiloaded languages. If $\mathscr{L}_1$ and $\mathscr{L}_2$ are families of languages, the arrow $\mathscr{L}_1 \to \mathscr{L}_2$ is supposed to be read as a proper inclusion $\mathscr{L}_1 \subsetneq \mathscr{L}_2$. A dotted line between two families of languages indicates that the families are incomparable.

Relations between various families of equiloaded languages have also been a subject of our study. We have proved numerous results on inclusions, strict inclusions, identities and incomparability relations between the families of equiloaded languages. We depict some of them in diagrams. In Figure Concl.1, the diagram representing the relations between the families of strictly transition-$\mathcal{S}$-equiloaded languages is shown. The diagram in Figure Concl.2 represents the relations between the analogous families of strictly state-$\mathcal{S}$-equiloaded languages. Finally, the diagram in Figure Concl.3 shows the relations between the families of transition-equiloaded DFA-languages and DFAε-languages.

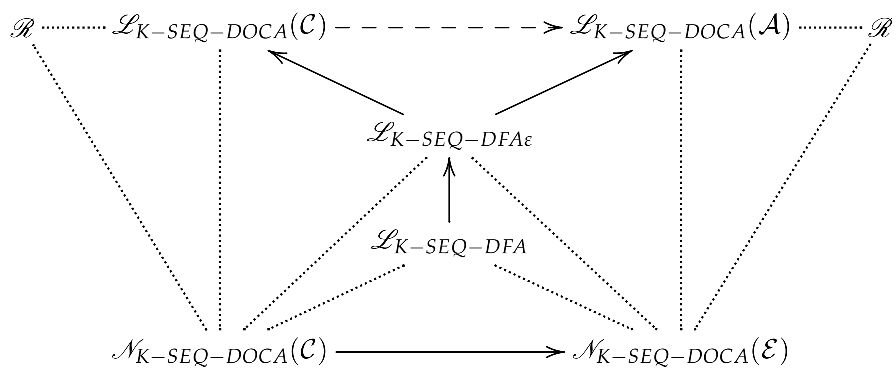


Figure Concl.2: The diagram of relations between various families of strictly state-$\mathcal{S}$-equiloaded languages. If $\mathscr{L}_1$ and $\mathscr{L}_2$ are families of languages, an arrow $\mathscr{L}_1 \rightarrow \mathscr{L}_2$ is supposed to be read as a proper inclusion $\mathscr{L}_1 \subsetneq \mathscr{L}_2$. A dashed arrow $\mathscr{L}_1 \dashrightarrow \mathscr{L}_2$ is an inclusion $\mathscr{L}_1 \subseteq \mathscr{L}_2$ with a strict inclusion being open. A dotted line between two families of languages indicates that the families are incomparable.

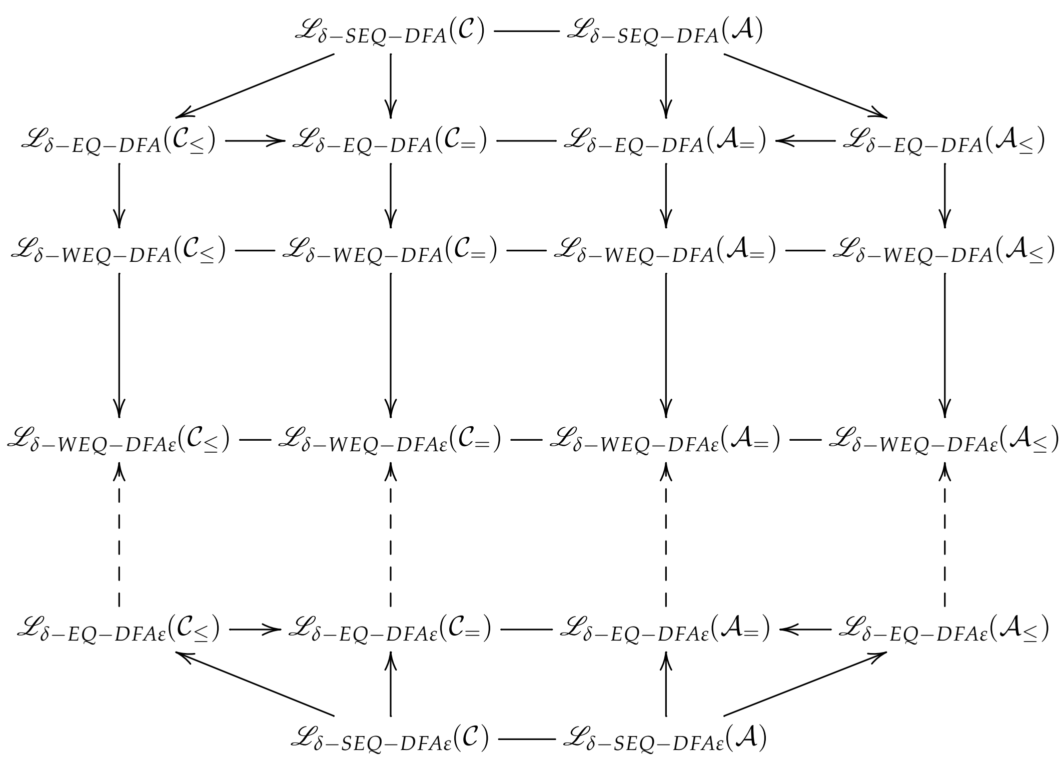


Figure Concl.3: The diagram of relations between various families of transition-equiloaded DFA(ε)-languages. If $\mathscr{L}_1$ and $\mathscr{L}_2$ are families of languages, an arrow $\mathscr{L}_1 \rightarrow \mathscr{L}_2$ is supposed to be read as a proper inclusion $\mathscr{L}_1 \subsetneq \mathscr{L}_2$. A dashed arrow $\mathscr{L}_1 \dashrightarrow \mathscr{L}_2$ is an inclusion $\mathscr{L}_1 \subseteq \mathscr{L}_2$ with a strict inclusion being open. A solid line between two families of languages indicates that the families are equal.

We have also studied the closure properties of several families of equiloaded languages. We summarize our results in Table Concl.2. The families of languages, for which we have not studied the closure properties yet, are omitted from the table. Some of the closure properties presented in this table have already been proved earlier – the closure properties of $\mathscr{L}_{K-SEQ-DFA}$ and $\mathscr{L}_{K-EQ-DFA}(\mathcal{A}_=)$ are due to [26] and [27], the closure properties of the families $\mathscr{L}_{\delta-SEQ-DFA}$, $\mathscr{L}_{\delta-EQ-DFA}(\mathcal{A}_=)$ and $\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{A}_=)$ are due to [25].

	$\cdot$	$\cup$	$\cap$	C	$*$	$+$	R	h	h^{-1}
$\mathscr{L}_{K-SEQ-DFA}$	No	No	Yes	No	No	No	No	No	No
$\mathscr{L}_{K-SEQ-DFA\varepsilon}$	No	No	Yes	No	No	No	No	No	No
$\mathscr{L}_{\delta-SEQ-DFA}$	No	No	Yes	No	No	No	No	No	No
$\mathscr{L}_{\delta-SEQ-DFA\varepsilon}$	No	No	Yes	No	No	No	No	Yes	No
$\mathscr{L}_{K-EQ-DFA}(\mathcal{A}_=)$	No	No	No	No	?	?	No	No	No
$\mathscr{L}_{\delta-EQ-DFA}(\mathcal{C}_=)$	No	No	No	No	No	No	No	No	No
$\mathscr{L}_{\delta-EQ-DFA}(\mathcal{A}_=)$	No	No	No	No	No	No	No	No	No
$\mathscr{L}_{\delta-EQ-DFA}(\mathcal{C}_\leq)$	No	No	No	No	No	No	No	No	No
$\mathscr{L}_{\delta-EQ-DFA}(\mathcal{A}_\leq)$	No	No	No	No	No	No	No	No	No
$\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{C}_=)$	No	No	No	No	?	?	?	?	No
$\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{A}_=)$	No	No	No	No	?	?	?	?	No
$\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{C}_\leq)$	No	No	No	No	?	?	?	?	No
$\mathscr{L}_{\delta-EQ-DFA\varepsilon}(\mathcal{A}_\leq)$	No	No	No	No	?	?	?	?	No
$\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{C}_=)$	No	No	No	No	No	No	No	No	No
$\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{A}_=)$	No	No	No	No	No	No	No	No	No
$\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{C}_\leq)$	No	No	No	No	No	No	No	No	No
$\mathscr{L}_{\delta-WEQ-DFA}(\mathcal{A}_\leq)$	No	No	No	No	No	No	No	No	No
$\mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{C}_=)$	No	No	No	No	?	?	?	?	No
$\mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{A}_=)$	No	No	No	No	?	?	?	?	No
$\mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{C}_\leq)$	No	No	No	No	?	?	?	?	No
$\mathscr{L}_{\delta-WEQ-DFA\varepsilon}(\mathcal{A}_\leq)$	No	No	No	No	?	?	?	?	No
$\mathscr{L}_{K-SEQ-DOCA}(\mathcal{C})$	No	?	?	No	?	?	No	?	No
$\mathscr{L}_{K-SEQ-DOCA}(\mathcal{A})$	No	?	?	No	?	?	No	?	No
$\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{C})$	No	No	?	No	No	No	No	?	No
$\mathscr{L}_{\delta-SEQ-DOCA}(\mathcal{A})$	No	No	?	No	No	No	No	?	No
$\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{C})$	No	No	?	No	No	No	No	?	No
$\mathscr{N}_{\delta-SEQ-DOCA}(\mathcal{E})$	No	No	?	No	No	No	No	?	No

Table Concl.2: The summary of closure properties of various families of equiloaded languages that have been proved up to now.

In our future research, we plan to focus mainly on the families of $\mathcal{S}$-equiloaded and weakly $\mathcal{S}$-equiloaded DOCA and to extend the theory also to some models of computation higher than DOCA. However, also some interesting open problems on equiloaded DFA(ε) and strictly $\mathcal{S}$-equiloaded DOCA have arisen in this report that may be as well a subject of a fruitful future research.

Bibliography

[1] BENVENUTI, L. – FARINA, L. 2004. Eigenvalue Regions for Positive Systems. In *Systems & Control Letters*. 2004, vol. 51, no. 3–4, pp. 325–330.

[2] BERMAN, A. – PLEMMONS, R. J. 1994. *Nonnegative Matrices in the Mathematical Sciences*. Philadelphia : SIAM, 1994. ISBN 0898713218.

[3] BRÉMAUD, P. 1999. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. New York : Springer, 1999. ISBN 0-387-98509-3.

[4] BOYD, S. – VANDENBERGHE, L. 2004. *Convex Optimization*. Cambridge : Cambridge University Press, 2004. ISBN 0-521-83378-7.

[5] BOYLE, M. *Notes on the Perron-Frobenius Theory of Nonnegative Matrices* [online]. Available online: `yaroslavvb.com/papers/boyle-notes.pdf`.

[6] CODDINGTON, E. A. – LEVINSON, N. 1987. *Theory of Ordinary Differential Equations* 9th ed. New Delhi : McGraw-Hill, 1987. ISBN 0898747554.

[7] COLLATZ, L. 1942. Einschließungssatz für die charakteristischen Zahlen von Matrizen. In *Mathematische Zeitschrift*. 1942, vol. 48, no. 1, pp. 221–226 (in German).

[8] CORMEN, T. H. – LEISERSON, C. E. – RIVEST, R. L. – STEIN, C. 2001. *Introduction to Algorithms* 2nd ed. Cambridge, Boston : MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7.

[9] DIESTEL, R. 2005. *Graph Theory*. Heidelberg : Springer, 2005. ISBN 978-3-642-14278-9.

[10] ELAYDI, S. 2005. *An Introduction to Difference Equations*. New York : Springer, 2005. ISBN 1441920013.

[11] FRANCIS, J. G. F. 1961. The QR Transformation, I. In *The Computer Journal*. 1961, vol. 4, no. 3, pp. 265–271.

[12] FRANCIS, J. G. F. 1962. The QR Transformation, II. In *The Computer Journal*. 1962, vol. 4, no. 4, pp. 332–345.

[13] FROBENIUS, G. 1912. Über Matrizen aus nicht negativen Elementen. In *Sitzungsberichte Königlich Preussichen Akademie der Wissenschaft*. 1912, pp. 456–477 (in German).

[14] GINSBURG, S. 1975. *Algebraic and Automata-Theoretic Properties of Formal Languages*. Amsterdam : North-Holland, 1975. ISBN 0720425069.

[15] GOLUB, G. H. – VAN LOAN, C. F. 1996. *Matrix Computations* 3rd ed. Baltimore : The John Hopkins University Press, 1996. ISBN 0-8018-5413-X.

[16] GROSS, J. L. – YELLEN, J. 2003. *Handbook of Graph Theory*. Abingdon : CRC Press, 2003. ISBN 1-58488-090-2.

[17] HARVILLE, D. A. 1997. *Matrix Algebra from a Statistician's Perspective* 1st ed. New York : Springer-Verlag, 1997. ISBN 0-387-94978-X.

[18] HASSELBLATT, B. – KATOK, A (eds). 2002. *Handbook of Dynamical Systems, Volume 1A*. Amsterdam : North-Holland, 2002. ISBN 0-444-82669-6.

[19] HOPCROFT, J. E. – MOTWANI, R. – ULLMAN, J. D. 2001. *Introduction to Automata Theory, Languages, and Computation* 2nd ed. Reading : Addison-Wesley, 2001. ISBN 0-201-44124-1.

[20] HOPCROFT, J. E. – ULLMAN, J. D. 1967. An Approach to a Unified Theory of Automata. In *Bell System Technical Journal*. 1967, vol. 46, no. 8, pp. 1793–1829.

[21] HOPCROFT, J. E. – ULLMAN, J. D. 1967. An Approach to a Unified Theory of Automata. In *Proceedings of the 8th Annual Symposium on Switching and Automata Theory (SWAT 1967)*. 1967, pp. 140–147.

[22] KALMAN, D. 1984. The Generalized Vandermonde Matrix. In *Mathematics Magazine*. 1984, vol. 57, no. 1, pp. 15–21.

[23] KARPELEVICH, F. I. 1951. O kharakteristicheskikh kornyakh matrits s neotritsatel'nymi elementami. In *Izvestiya Akademii Nauk SSSR. Seriya Matematicheskaya*. 1951, vol. 15, pp. 361–383 (in Russian).

[24] KELLEY, W. G. – PETERSON A. C. 2001. *Difference Equations - An Introduction with Applications* 2nd ed. San Diego : Academic Press, 2001. ISBN 0-12-403330-X.

[25] KOSTOLÁNYI, P. 2011. *Rovnomerné využívanie prechodov v konečných automatoch* : Bachelor's Thesis. Bratislava : Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava, 2011 (in Slovak).

[26] KOVÁČ, I. 2010. *Equiloaded Automata* : Master's Thesis. Bratislava : Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava, 2010.

[27] KOVÁČ, I. 2008. *O využívaní stavov v konečných automatoch* : Bachelor's Thesis. Bratislava : Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava, 2008 (in Slovak).

[28] KUBLANOVSKAYA, V. 1963. On some algorithms for the solution of the complete eigenvalue problem. In *USSR Computational Mathematics and Mathematical Physics*. 1963, vol. 1, no. 3, pp. 637–657.

[29] LANG, S. 1996. *Undergraduate Analysis* 2nd ed. New York : Springer, 1996. ISBN 0387948414.

[30] MEYER, C. D. 2001. *Matrix Analysis and Applied Linear Algebra*. Philadelphia : SIAM, 2001. ISBN 0898714540.

[31] MINC, H. 1988. *Nonnegative Matrices*. New York : Wiley, 1988. ISBN 0471839663.

[32] NERODE, A. 1958. Linear Automaton Transformations. In *Proceedings of the American Mathematical Society*. ISSN 0002-9939, 1958, vol. 9, no. 4, pp. 541 – 544.

[33] PERRON, O. 1907. Zur Theorie der Matrices. In *Mathematische Annalen*. 1907, vol. 64, no. 2, pp. 248–263 (in German).

[34] ROVAN, B. – FORIŠEK, M. 2009. *Formálne jazyky a automaty* [online]. Available online: `http://foja.dcs.fmph.uniba.sk/materialy/skripta.pdf` (in Slovak).

[35] SIMMONS, G. F. 1963. *Introduction to Topology and Modern Analysis*. New York : McGraw-Hill, 1963. ISBN 0070573891.

[36] STRANG, G. 2007. *Computational Science and Engineering*. Wellesley : Wellesley-Cambridge Press, 2007. ISBN 0-9614088-1-2.

[37] TREIL, S. 2009. *Linear Algebra Done Wrong* [online]. Available online: `http://www.math.brown.edu/~treil/papers/LADW/LADW.pdf`.

[38] VALIANT, L. G. – PATERSON, M. S. 1975. Deterministic One-Counter Automata. In *Journal of Computer and System Sciences*. 1975, vol. 20, no. 3, pp. 340 – 350.

[39] VARGA, R. S. 2000. *Matrix Iterative Analysis* 2nd ed. Berlin : Springer, 2000. ISBN 978-3-540-66321-8.

[40] WIELANDT, H. 1950. Unzerlegbare, nicht negative Matrizen. In *Mathematische Zeitschrift*. 1950, vol. 52, no. 1., pp. 642–648 (in German).

[41] ZLATOŠ, P. 2011. *Lineárna algebra a geometria*. Bratislava : Albert Marenčin PT, 2011. ISBN 978-80-8114-111-9 (in Slovak).

Appendix A

Mathematical Preliminaries

In this appendix, we shall briefly review some of the mathematical concepts used in this report. We shall concentrate only on (from the perspective of theoretical computer science) more advanced topics. That is, we assume that the reader is familiar with some of the basic mathematics, most importantly linear algebra, discrete mathematics, and calculus.

A.1 Vandermonde and Generalized Vandermonde Matrices

In this section, we shall define *Vandermonde matrices* and *generalized Vandermonde matrices* and shall state the formulas for their determinants (the proof of the formula for a generalized Vandermonde determinant will be omitted). The results obtained in this section shall be used in Section A.2 dealing with proving linear independence of sequences.

Definition A.1.1 Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ in $\mathbb{C}$ be complex numbers. A matrix

$$V(\alpha_1, \alpha_2, \ldots, \alpha_n) = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{n-1} & \alpha_2^{n-1} & \ldots & \alpha_n^{n-1} \end{pmatrix}$$

is said to be a *Vandermonde matrix*. The determinant of a Vandermonde matrix is said to be a *Vandermonde determinant*.

Some authors define a Vandermonde matrix to be the transpose of this matrix (see, e.g., [36]). In this report we follow the definition used, e.g., in [10], or [15].[1] However, this ambiguity does not have any effect on the value of a Vandermonde determinant and in most applications of Vandermonde matrices, we are concerned solely by the value of the determinant. The choice of one of these definitions can be therefore considered to be only a matter of convenience. The formula for a Vandermonde determinant is being proved in the following theorem.

Theorem A.1.2 Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ in $\mathbb{C}$ be complex numbers, let $V(\alpha_1, \alpha_2, \ldots, \alpha_n)$ be the corresponding Vandermonde matrix. Then the value of the corresponding Vandermonde determinant is

$$|V(\alpha_1, \alpha_2, \ldots, \alpha_n)| = \prod_{1 \le i < j \le n} (\alpha_j - \alpha_i).$$

Proof. By induction on n (we shall follow a proof presented in [22]).

1. If $n = 1$, then the determinant $|V(\alpha_1)|$ is clearly 1. That is, the theorem holds.

[1] However, in [15], a Vandermonde matrix is defined only for real entries.

2. Let us suppose, that the theorem holds for $n = k$. We shall prove that it holds for $n = k + 1$. Let us denote
$$g(t) = |V(\alpha_1, \alpha_2, \ldots, \alpha_k, t)|,$$
that is clearly a polynomial of degree k with roots $\alpha_1, \ldots, \alpha_k$. The leading coefficient of the polynomial $g(t)$ is obtained by the Laplace expansion along the $(k+1)$-st column, that is the leading coefficient is $|V(\alpha_1, \alpha_2, \ldots, \alpha_k)|$. Thus, we have
$$g(t) = |V(\alpha_1, \alpha_2, \ldots, \alpha_k)| \cdot \prod_{i=1}^{k}(t - \alpha_i).$$
Using the induction hypothesis, we get
$$g(t) = \prod_{1 \le i < j \le k} (\alpha_j - \alpha_i) \cdot \prod_{i=1}^{k}(t - \alpha_i),$$
therefore for $t = \alpha_{k+1}$ we get
$$|V(\alpha_1, \alpha_2, \ldots, \alpha_k, \alpha_{k+1})| = \prod_{1 \le i < j \le k+1} (\alpha_j - \alpha_i),$$
that is, the theorem holds also for $n = k + 1$.

The theorem is thus proven. □

Now, we shall define a *generalized Vandermonde matrix* [22]. Let us note, that for $k = n$, and $n_1 = n_2 = \ldots = n_k = 1$, the generalized Vandermonde matrix is exactly the above-defined Vandermonde matrix.

Definition A.1.3 Let $\alpha_1, \alpha_2, \ldots, \alpha_k$ in $\mathbb{C}$ be arbitrary complex numbers, let $n_1, n_2, \ldots, n_k$ in $\mathbb{N}$ be nonnegative integers, such that $\sum_{i=1}^{k} n_i = n$ for some positive n in $\mathbb{N}$. The *generalized Vandermonde Matrix* $V(\alpha_1, \ldots, \alpha_k; n_1, \ldots, n_k)$ is defined to be a block matrix
$$V(\alpha_1, \ldots, \alpha_k; n_1, \ldots, n_k) = (R_1, R_2, \ldots, R_k),$$
where, for $i = 1, \ldots, k$, R_i is a matrix of type $n \times n_i$ with columns $\mathbf{c}_{i,1}, \ldots, \mathbf{c}_{i,n_i}$ defined by
$$\mathbf{c}_{i,j} = \frac{1}{(j-1)!}\mathbf{f}^{(j-1)}(\alpha_i)$$
for $j = 1, \ldots, n_i$, where $\mathbf{f}(t)$ is defined to be a column vector
$$\mathbf{f}(t) = (1, t, \ldots, t^{n-1})^T,$$
and $\mathbf{f}^{(k)}(t)$ is its k-th derivative (i.e., a column vector consisting of k-th derivatives of entries of $\mathbf{f}(t)$).

The determinant of a generalized Vandermonde matrix is said to be a *generalized Vandermonde determinant*.

Next, we shall state a formula for a generalized Vandermonde determinant. However, since the proof is rather technical, we shall not present it in this report.

Theorem A.1.4 Let $\alpha_1, \alpha_2, \ldots, \alpha_k$ in $\mathbb{C}$ be complex numbers, let $n_1, n_2, \ldots, n_k$ in $\mathbb{N}$ be nonnegative integers, such that $\sum_{i=1}^{k} n_i = n$ for some positive n in $\mathbb{N}$. Then the value of the corresponding generalized Vandermonde determinant is given by
$$|V(\alpha_1, \ldots, \alpha_k; n_1, \ldots, n_k)| = \prod_{1 \le i < j \le k} (\alpha_j - \alpha_i)^{n_i n_j}.$$

Proof. A sketch of several possible proofs of this theorem can be found in [22]. Another sketch of a proof can be found in [10]. □

A.2 Proving Linear Independence of Sequences

In this section, we shall provide basic information about proving linear independence of sequences. More concretely, we shall introduce the concept of the matrix of Casorati, and state a theorem about linear independence of sequences of certain specific form. The linear independence of these special sequences is used in our examinations in Chapter 2. The results can be also used to build up the theory of systems of linear OΔEs. However, in Section A.3 dealing with these matters, we use a different, faster approach.

First, we shall introduce the concept of the *matrix of Casorati* and its determinant, referred to as the *Casoratian*. These concepts are closely related to the *matrix of Wronski* and the *Wronskian*, playing a crucial role in the theory of linear differential equations (see, e.g., [6]).

Definition A.2.1 Let $x_1, x_2, \ldots, x_m : \mathbb{N} \to \mathbb{C}$ be given complex sequences. The *matrix of Casorati* for sequences $x_1, x_2, \ldots, x_m$ is the matrix

$$W(n) = \begin{pmatrix} x_1(n) & x_2(n) & \ldots & x_m(n) \\ x_1(n+1) & x_2(n+1) & \ldots & x_m(n+1) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(n+m-1) & x_2(n+m-1) & \ldots & x_m(n+m-1) \end{pmatrix}.$$

The determinant

$$w(n) = |W(n)|$$

of the matrix of Casorati is called the *Casoratian*.

In the following theorem we shall prove that if a Casoratian of a given set of complex sequences is non-zero, then the set is linearly independent.

Theorem A.2.2 Let $x_1, x_2, \ldots, x_m : \mathbb{N} \to \mathbb{C}$ be given complex sequences. Let $w(n)$ be a Casoratian for these sequences and for some n in $\mathbb{N}$. If $w(n) \neq 0$, then the sequences $x_1, x_2, \ldots, x_m$ are linearly independent.

Proof. For the purpose of contradiction, let us suppose that $w(n) \neq 0$ and sequences $x_1, x_2, \ldots, x_m$ are linearly dependent, that is not-all-zero constants $c_1, c_2, \ldots, c_m$ in $\mathbb{C}$ exist such that for all non-negative integers k in $\mathbb{N}$

$$c_1 x_1(k) + c_2 x_2(k) + \ldots + c_m x_m(k) = 0. \tag{A.1}$$

Since the equation (A.1) holds for all k in $\mathbb{N}$, it holds also for $k = n, n+1, \ldots, n+m-1$. From that we have

$$\begin{aligned} c_1 x_1(n) + c_2 x_2(n) + \ldots + c_m x_m(n) &= 0, \\ c_1 x_1(n+1) + c_2 x_2(n+1) + \ldots + c_m x_m(n+1) &= 0, \\ &\vdots \\ c_1 x_1(n+m-1) + c_2 x_2(n+m-1) + \ldots + c_m x_m(n+m-1) &= 0. \end{aligned}$$

That is, we have a homogeneous system of linear equations with unknowns $c_1, c_2, \ldots, c_m$, which can be written in a matrix form as

$$\begin{pmatrix} x_1(n) & x_2(n) & \ldots & x_m(n) \\ x_1(n+1) & x_2(n+1) & \ldots & x_m(n+1) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(n+m-1) & x_2(n+m-1) & \ldots & x_m(n+m-1) \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

i.e.,

$$W(n) \cdot \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Since $w(n) \neq 0$, the matrix of the system $W(n)$ is nonsingular and a system has only one trivial solution $(c_1, c_2, \dots, c_m) = (0, 0, \dots, 0)$, and that contradicts (A.1). □

Thus, the Casoratian can be used as a powerful tool for proving linear independence of sequences. In the following theorem, we shall use the Casoratian to prove the linear independence of sequences of certain specific form.

Theorem A.2.3 Let $z_1, z_2, \dots, z_k$ in $\mathbb{C} \setminus \{0\}$ be distinct nonzero complex numbers, for some k in $\mathbb{N}$. Let $m_1, m_2, \dots, m_k$ be nonnegative integers. Then the sequences

$$\begin{gathered} \{z_1^n\}_{n=0}^{\infty}, \{n \cdot z_1^n\}_{n=0}^{\infty}, \dots, \{n^{m_1} \cdot z_1^n\}_{n=0}^{\infty}, \\ \{z_2^n\}_{n=0}^{\infty}, \{n \cdot z_2^n\}_{n=0}^{\infty}, \dots, \{n^{m_2} \cdot z_2^n\}_{n=0}^{\infty}, \\ \vdots \\ \{z_k^n\}_{n=0}^{\infty}, \{n \cdot z_k^n\}_{n=0}^{\infty}, \dots, \{n^{m_k} \cdot z_k^n\}_{n=0}^{\infty} \end{gathered}$$

are linearly independent.

Proof. For the purpose of contradiction, let us suppose that the sequences are linearly dependent. That is, constants $c_{i,j}$ in $\mathbb{C}$ exist for $i = 1, \dots, k, j = 0, \dots, m_i$, such that

$$\sum_{i=1}^{k} \sum_{j=0}^{m_i} c_{i,j} \cdot n^j \cdot z_i^n = 0.$$

Now, since $\binom{n}{j}$ is a polynomial of degree j, it is clear that for each j in $\mathbb{N}$ constants $d_0, d_1, \dots, d_j$ in $\mathbb{C}$ exist such that

$$n^j = \sum_{i=0}^{j} d_i \cdot \binom{n}{j}.$$

Thus, if the sequences

$$\begin{gathered} \{z_1^n\}_{n=0}^{\infty}, \{n \cdot z_1^n\}_{n=0}^{\infty}, \dots, \{n^{m_1} \cdot z_1^n\}_{n=0}^{\infty}, \\ \{z_2^n\}_{n=0}^{\infty}, \{n \cdot z_2^n\}_{n=0}^{\infty}, \dots, \{n^{m_2} \cdot z_2^n\}_{n=0}^{\infty}, \\ \vdots \\ \{z_k^n\}_{n=0}^{\infty}, \{n \cdot z_k^n\}_{n=0}^{\infty}, \dots, \{n^{m_k} \cdot z_k^n\}_{n=0}^{\infty} \end{gathered}$$

are linearly dependent, then also the sequences

$$\begin{gathered} \left\{\binom{n}{0} \cdot z_1^n\right\}_{n=0}^{\infty}, \left\{\binom{n}{1} \cdot z_1^n\right\}_{n=0}^{\infty}, \dots, \left\{\binom{n}{m_1} \cdot z_1^n\right\}_{n=0}^{\infty}, \\ \left\{\binom{n}{0} \cdot z_2^n\right\}_{n=0}^{\infty}, \left\{\binom{n}{1} \cdot z_2^n\right\}_{n=0}^{\infty}, \dots, \left\{\binom{n}{m_2} \cdot z_2^n\right\}_{n=0}^{\infty}, \\ \vdots \\ \left\{\binom{n}{0} \cdot z_k^n\right\}_{n=0}^{\infty}, \left\{\binom{n}{1} \cdot z_k^n\right\}_{n=0}^{\infty}, \dots, \left\{\binom{n}{m_k} \cdot z_k^n\right\}_{n=0}^{\infty} \end{gathered}$$

are linearly dependent. Moreover, since $z_1, \ldots, z_k$ are nonzero, $z_i^n = z^j \cdot z^{n-j}$ for all j in $\mathbb{N}$ and $i = 1, \ldots, k$, where z^j is a constant term. That is, the sequences

$$\begin{array}{c}
\left\{\binom{n}{0} \cdot z_1^n\right\}_{n=0}^{\infty}, \left\{\binom{n}{1} \cdot z_1^{n-1}\right\}_{n=0}^{\infty}, \ldots, \left\{\binom{n}{m_1} \cdot z_1^{n-m_1}\right\}_{n=0}^{\infty}, \\
\left\{\binom{n}{0} \cdot z_2^n\right\}_{n=0}^{\infty}, \left\{\binom{n}{1} \cdot z_2^{n-1}\right\}_{n=0}^{\infty}, \ldots, \left\{\binom{n}{m_2} \cdot z_2^{n-m_2}\right\}_{n=0}^{\infty}, \\
\vdots \\
\left\{\binom{n}{0} \cdot z_k^n\right\}_{n=0}^{\infty}, \left\{\binom{n}{1} \cdot z_k^{n-1}\right\}_{n=0}^{\infty}, \ldots, \left\{\binom{n}{m_k} \cdot z_k^{n-m_k}\right\}_{n=0}^{\infty}
\end{array}$$

are also linearly dependent. However, the matrix of Casorati for these sequences is clearly the generalized Vandermonde matrix $V(z_1, \ldots, z_k; m_1 + 1, \ldots, m_k + 1)$, and therefore the Casoratian for these sequences has a value of the generalized Vandermonde determinant

$$w(n) = \prod_{1 \leq i < j \leq k} (z_j - z_i)^{(m_i+1)(m_j+1)}.$$

However, since $z_1, \ldots, z_k$ are distinct, the Casoratian $w(n)$ is clearly nonzero, and the sequences are linearly independent, i.e., a contradiction. □

A.3 Systems of Linear OΔEs

In this section, we shall discuss systems of first-order linear ordinary difference equations (OΔEs, i.e., recurrences)[2] and present a linear-algebraic method of solving initial value problems for homogeneous systems of first-order linear OΔEs with constant coefficients. The presented method can also be easily extended to nonhomogeneous systems. For a more comprehensive treatment, see, e.g., [10], or [24].

Definition A.3.1 An *initial value problem for a system of first-order linear OΔEs* is a problem of determining m unknown functions[3]

$$x_1, x_2, \ldots, x_m : \mathbb{N} \to \mathbb{C},$$

satisfying

$$\begin{aligned}
x_1(n) &= a_{1,1}(n)x_1(n-1) + a_{1,2}(n)x_2(n-1) + \ldots + a_{1,m}(n)x_m(n-1) + f_1(n) \\
x_2(n) &= a_{2,1}(n)x_1(n-1) + a_{2,2}(n)x_2(n-1) + \ldots + a_{2,m}(n)x_m(n-1) + f_2(n) \\
&\vdots \\
x_m(n) &= a_{m,1}(n)x_1(n-1) + a_{m,2}(n)x_2(n-1) + \ldots + a_{m,m}(n)x_m(n-1) + f_m(n)
\end{aligned}$$

for $n \geq 1$ with initial conditions

$$x_1(0) = C_1, x_2(0) = C_2, \ldots, x_m(0) = C_m,$$

where $a_{i,j} : \mathbb{N}^+ \to \mathbb{C}$ (with $1 \leq i, j \leq m$) and $f_i : \mathbb{N}^+ \to \mathbb{C}$ (with $1 \leq i \leq m$) are functions and $C_1, C_2, \ldots, C_m$ in $\mathbb{C}$ are complex constants.

[2]Not to be confused with differential equations. Although this need not be always true, in this report we use the terms *difference equation* and *recurrence* as synonyms. However, we shall prefer the term *difference equation*.

[3]From a strictly mathematical viewpoint, the term *sequence* is more appropriate. However, in computer science it is common to call sequences of this type functions. In this section, we shall therefore be switching between both terms.

The system can be therefore expressed in a vector-matrix notation as

$$\mathbf{x}_n = A_n \cdot \mathbf{x}_{n-1} + \mathbf{f}_n, \qquad n \geq 1,$$

where

$$\mathbf{x}_n = (x_1(n), x_2(n), \ldots, x_m(n))^T$$

is the n-th vector of unknown functions,

$$A_n = \begin{pmatrix} a_{1,1}(n) & a_{1,2}(n) & \ldots & a_{1,m}(n) \\ a_{2,1}(n) & a_{2,2}(n) & \ldots & a_{2,m}(n) \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1}(n) & a_{m,2}(n) & \ldots & a_{m,m}(n) \end{pmatrix}$$

is the n-th matrix of the system,

$$\mathbf{f}_n = (f_1(n), f_2(n), \ldots, f_m(n))^T$$

is the n-th vector of known functions f_i, and where the vector of initial conditions

$$\mathbf{x}_0 = (C_1, C_2, \ldots, C_m) \in \mathbb{C}^m$$

is given. A sequence $\{\mathbf{x}_n\}_{n=0}^{\infty}$ with values in $\mathbb{C}^m$ is said to be a *solution* to the initial value problem, if it satisfies the conditions above.

Moreover, we shall say that the system is *homogeneous*, if the identities

$$f_i(n) \equiv 0, \qquad i = 1, \ldots, m$$

hold. Otherwise, we shall say that the system is *nonhomogeneous*. Finally, if constants $c_{i,j}$ in $\mathbb{C}, 1 \leq i, j \leq m$ exist, such that

$$a_{i,j}(n) \equiv c_{i,j}, \qquad i, j = 1, \ldots, m,$$

we say that the system has *constant coefficients*.[4] Thus, any *homogeneous system of first-order linear OΔEs with constant coefficients* can be written as

$$\mathbf{x}_n = A \cdot \mathbf{x}_{n-1}, \qquad n \geq 1,$$

where $\mathbf{x}_n$ is the n-th vector of unknown functions, and A is a matrix. Initial conditions are given by

$$\mathbf{x}_0 = (C_1, C_2, \ldots, C_m)^T.$$

In what follows, we shall present a method based on linear algebra that enables us to solve any initial value problem for homogeneous systems of first-order linear OΔEs with constant coefficients (to be more precise, in this report we shall be interested only in finding the closed form for $x_1(n)$ instead of $\mathbf{x}_n$). Let us begin with the theorem that ensures that every such initial value problem has exactly one solution.

Theorem A.3.2 The initial value problem for homogeneous system of first-order linear OΔEs with constant coefficients

$$\mathbf{x}_n = A \cdot \mathbf{x}_{n-1}, \qquad n \geq 1 \tag{A.2}$$

has a unique solution for each given $\mathbf{x}_0$.

[4] Systems with constant coefficients are also referred to as *autonomous systems*.

Proof. Let us define $\mathbf{x}_n$ for $n \geq 1$ by

$$\mathbf{x}_n = A^n \cdot \mathbf{x}_0.$$

Clearly, this is a solution to the initial value problem.

The uniqueness of the solution stems from the fact that for every solution $\mathbf{y} = \{\mathbf{y}_0, \mathbf{y}_1, \ldots\}$, the equation (A.2) implies

$$\mathbf{y}_n = A^n \cdot \mathbf{y}_0 = A^n \cdot \mathbf{x}_0.$$

However, $\mathbf{x} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots\}$ is clearly the only sequence of vectors satisfying this property. □

In what follows, we shall make a strict distinction between systems and initial value problems in the sense that for systems there is no initial vector $\mathbf{x}_0$ given.

By a *particular solution* of a given system, we shall understand any solution of some initial value problem for that system. By the *general solution* of a system, we shall understand the set of all particular solutions of the system. In what follows, we shall show that a general solution of a system is in fact a linear space – that is, for each particular solution, any scalar multiple of that solution is also a particular solution, and for any two particular solutions, their sum is also a particular solution. More concisely, any linear combination of two particular solutions is also a particular solution.

Theorem A.3.3 Let $\mathbf{u} = \{\mathbf{u}_0, \mathbf{u}_1, \ldots\}$, $\mathbf{v} = \{\mathbf{v}_0, \mathbf{v}_1, \ldots\}$ be particular solutions of a homogeneous system of first-order linear OΔEs with constant coefficients,

$$\mathbf{x}_n = A \cdot \mathbf{x}_{n-1}, \qquad n \geq 1, \tag{A.3}$$

and let α, β in $\mathbb{C}$ be arbitrary complex numbers. Then

$$\mathbf{z} = \alpha\mathbf{u} + \beta\mathbf{v} = \{\alpha\mathbf{u}_0 + \beta\mathbf{v}_0, \alpha\mathbf{u}_1 + \beta\mathbf{v}_1, \ldots\}$$

is a particular solution of the system (A.3).

Proof. The particular solution $\mathbf{u}$ is a solution to an initial value problem for system (A.3), where the initial conditions are given by $\mathbf{u}_0$. Similarly, $\mathbf{v}$ is a solution to an initial value problem for system (A.3), where the initial conditions are given by $\mathbf{v}_0$.

Let us now consider an initial value problem for system (A.3), where the initial conditions are given by $\alpha\mathbf{u}_0 + \beta\mathbf{v}_0$. By Theorem A.3.2, this initial value problem has a unique solution, say $\mathbf{w} = \{\mathbf{w}_0, \mathbf{w}_1, \ldots\}$. We shall prove that $\mathbf{w} = \mathbf{z}$.

Clearly, for $\mathbf{w}$ we have

$$\mathbf{w}_0 = \alpha\mathbf{u}_0 + \beta\mathbf{v}_0, \tag{A.4}$$

and

$$\mathbf{w}_n = A^n\mathbf{w}_0, \qquad n \geq 1. \tag{A.5}$$

By plugging the right side of (A.4) into (A.5), we get

$$\mathbf{w}_n = A^n(\alpha\mathbf{u}_0 + \beta\mathbf{v}_0) = \alpha A^n\mathbf{u}_0 + \beta A^n\mathbf{v}_0 = \alpha\mathbf{u}_n + \beta\mathbf{v}_n = \mathbf{z}_n, \qquad n \geq 1,$$

that is, $\mathbf{w} = \mathbf{z}$. Thus, we have proved that $\mathbf{z}$ is a solution to some initial value problem for (A.3), that is, $\mathbf{z}$ is a particular solution. □

The general solution of a given homogeneous system of first-order linear OΔEs with constant coefficients therefore forms a linear space.

There is more then one possible way to solve initial value problems for systems of our type. One frequently used approach is based on finding m linearly independent particular solutions (where m is the size of the matrix of the system) and proving that every system with the matrix of size m has a general solution of dimension at most m. Thus, the general solution is the set of all linear combinations of those m particular solutions. Each particular solution has a form $f(n)\mathbf{c}$, where $f : \mathbb{N} \to \mathbb{C}$ is a function, and $\mathbf{c} \in \mathbb{C}^m$ is a constant vector. Thus, the general solution is then expressed as a set of linear combinations

$$c^{(1)} f^{(1)}(n)\mathbf{c}^{(1)} + \ldots + c^{(m)} f^{(m)}(n)\mathbf{c}^{(m)},$$

where $f^{(1)}, \ldots, f^{(m)} : \mathbb{N} \to \mathbb{C}$ are fixed functions, $\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(m)}$ are fixed vectors, and $c^{(1)}, \ldots, c^{(m)}$ are variable coefficients. For a given initial value problem, it is possible to determine the variable coefficients by solving a system of linear equations. This approach is similar to the methods used in the theory of linear *differential* equations (see, e.g., [6]). For proving linear independence of functions occurring in the general solution, one needs the concepts of the matrix of Casorati (see Section A.2), and of the generalized Vandermonde matrix (see Section A.1).

In this report, we shall use a slightly different (though closely related) method of solving initial value problems. Instead of finding a general solution for $\mathbf{x}_n$, we shall find a "general solution" for each of its components $x_i(n)$, $i = 1, \ldots, m$. Thus, instead of m variable coefficients, we will have m^2 variable coefficients. However, the derivation of the method will be considerably simpler and we shall avoid the need of computing (possibly generalized) eigenvectors. Moreover, in this report we shall be always interested only in finding a solution for one single component of $\mathbf{x}_n$, so the number of variable coefficients to be determined will remain unchanged (i.e., m).

However, let us note that although the method to be presented works fine for the purpose of solving initial value problems, it is insufficient for the purpose of finding the general solution for $\mathbf{x}_n$ – for this purpose, the method sketched above is more suitable.

Theorem A.3.4 Let m in $\mathbb{N}^+$ be a positive integer, and A be an $m \times m$ matrix with distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_k$ with algebraic multiplicities $\alpha_1, \alpha_2, \ldots, \alpha_k, \alpha_1 + \alpha_2 + \ldots + \alpha_k = m$. Then, for $s = 1, \ldots, m$, the s-th component of the solution to the initial value problem

$$\mathbf{x}_n = A \cdot \mathbf{x}_{n-1}, \qquad n \geq 1 \tag{A.6}$$

with $\mathbf{x}_n$ denoting a column vector

$$\mathbf{x}_n = (x_1(n), x_2(n), \ldots, x_m(n))^T$$

and with initial conditions given by a column vector

$$\mathbf{x}_0 = (C_1, C_2, \ldots, C_m)^T \in \mathbb{C}^{m \times 1}$$

can be expressed in a form

$$x_s(n) = \sum_{i=1}^{k} \sum_{j=0}^{\alpha_i - 1} c_{i,j,s} \binom{n}{j} \lambda_i^{n-j} \tag{A.7}$$

for some constants $c_{i,j,s}$, $i = 1, \ldots, k$, $j = 0, \ldots, \alpha_i - 1$.[5]

Proof. It is clear that the equation

$$\begin{pmatrix} x_1(n) \\ x_2(n) \\ \vdots \\ x_m(n) \end{pmatrix} = A^n \cdot \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \end{pmatrix} \tag{A.8}$$

holds. By the well-known result of linear algebra (see, e.g., [41]), the matrix A can be decomposed into the form

$$A = P \cdot J \cdot P^{-1},$$

[5] In this result, we use these common conventions: $\binom{n}{j}\lambda^{n-j} = 0$ for $j > n$, and $0^0 = 1$.

where P is an invertible matrix, and J is a matrix in the Jordan canonical form, i.e., J is a block-diagonal matrix

$$J = \begin{pmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_r \end{pmatrix},$$

where r in $\mathbb{N}^+$ is a positive integer satisfying $k \le r \le m$, and for each $i = 1, \ldots, r$, J_i is a Jordan block corresponding to some eigenvalue[6] λ_j, $1 \le j \le k$, i.e., a square matrix of size $m_i \times m_i$ (for some nonnegative integer m_i in $\mathbb{N}$ satisfying $1 \le m_i \le \alpha_j$)[7] of the form

$$J_i = \begin{pmatrix} \lambda_j & 1 & & \\ & \lambda_j & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_j \end{pmatrix}.$$

According to basic results of linear algebra [41], the properties

$$J_i^n = \begin{pmatrix} \lambda_j^n & \binom{n}{1}\lambda_j^{n-1} & \ldots & \binom{n}{m_i-1}\lambda_j^{n-m_i+1} \\ 0 & \lambda_j^n & \ldots & \binom{n}{m_i-2}\lambda_j^{n-m_i+2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_j^n \end{pmatrix}, \qquad i = 1, \ldots, r,$$

and

$$J^n = \begin{pmatrix} J_1^n & & & \\ & J_2^n & & \\ & & \ddots & \\ & & & J_r^n \end{pmatrix}$$

hold. Moreover, it is clear that

$$A^n = P \cdot J^n \cdot P^{-1}. \tag{A.9}$$

Therefore, by plugging the right-hand side of (A.9) to (A.8), we obtain

$$\begin{pmatrix} x_1(n) \\ x_2(n) \\ \vdots \\ x_m(n) \end{pmatrix} = P \cdot J^n \cdot P^{-1} \cdot \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \end{pmatrix}.$$

However, since $C_1, \ldots, C_m$ are all constants, and since all of the entries of P and P^{-1} are constant as well, from the form of J^n it is obvious that the s-th entry of the left-hand side column vector has a form

$$x_s(n) = \sum_{i=1}^{k} \sum_{j=0}^{\alpha_i - 1} c_{i,j,s} \binom{n}{j} \lambda_i^{n-j} \tag{A.10}$$

for some suitable constants $c_{i,j,s}$, $i = 1, \ldots, k$, $j = 0, \ldots, \alpha_i - 1$ (the size of each Jordan block corresponding to eigenvalue λ_i is at most α_i). The theorem is proved. □

[6] There can be more than one Jordan block corresponding to the same eigenvalue.

[7] Moreover, $m_1 + m_2 + \ldots + m_r = m$. The number of blocks corresponding to a given eigenvalue is its geometric multiplicity. Each block corresponds to some subspace of dimension 1 of the eigenspace corresponding to the given eigenvalue and its size is given by the number of eigenvectors from this subspace.

Since $\binom{n}{j}$ is a polynomial of degree j,[8] it is clear that for $\lambda_i \neq 0$

$$\sum_{j=0}^{\alpha_i-1} c_{i,j,s} \binom{n}{j} \lambda_i^{n-j} = \sum_{j=0}^{\alpha_i-1} c_{i,j,s} \binom{n}{j} \frac{1}{\lambda_i^j} \lambda_i^n = p(n)\lambda_i^n, \tag{A.11}$$

where $p(n)$ is a polynomial of degree at most $\alpha_i - 1$. Thus, it seems that (A.7) can be significantly simplified. However, there is a problem with repeated zero eigenvalues, since in general it is impossible to write 0^{n-j} in a form $c \cdot 0^n$ for some constant c (as it was in the case of nonzero λ_i). However, $\binom{n}{j} 0^{n-j}$ can be expressed in the Iverson notation[9] as $[n = j]$. Thus, we have proved the following corollary:

Corollary A.3.5 Let m in $\mathbb{N}^+$ be a positive integer, and A be a $m \times m$ matrix with distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_k$ with algebraic multiplicities $\alpha_1, \alpha_2, \ldots, \alpha_k, \alpha_1 + \alpha_2 + \ldots + \alpha_k = m$. Let α be a multiplicity of the zero eigenvalue. Then, for $s = 1, \ldots, m$, the s-th component of the solution to the initial value problem

$$\mathbf{x}_n = A \cdot \mathbf{x}_{n-1}, \qquad n \geq 1 \tag{A.12}$$

with $\mathbf{x}_n$ denoting a column vector

$$\mathbf{x}_n = (x_1(n), x_2(n), \ldots, x_m(n))^T$$

and with initial conditions given by a column vector

$$\mathbf{x}_0 = (C_1, C_2, \ldots, C_m)^T \in \mathbb{C}^{m \times 1}$$

can be expressed in a form

$$x_s(n) = \sum_{\substack{i \\ \lambda_i \neq 0}} \sum_{j=0}^{\alpha_i-1} c_{i,j,s} \cdot n^j \lambda_i^n + \sum_{j=0}^{\alpha-1} c_{n=j,s} \cdot [n = j] \tag{A.13}$$

for some constants $c_{i,j,s}$, $i = 1, \ldots, k$, $j = 0, \ldots, \alpha_i - 1$ and $c_{n=j,s}$, $j = 0, \ldots, \alpha - 1$.

Thus, the method of solving initial value problems for homogeneous systems of first-order linear OΔEs with constant coefficients can be summarized as follows:

1. Write down the system in a matrix form $\mathbf{x}_n = A \cdot \mathbf{x}_{n-1}$.
2. Compute the eigenvalues of the matrix A.
3. The solution for each component of $\mathbf{x}_n$ has a form (A.13).
4. Determine the constants in (A.13) by solving a system of linear equations obtained from the initial conditions.

Let us note that this method can be easily turned into a numerical algorithm for solving these systems, with a time complexity constant with respect to n (i.e., dependent only on m). The only nontrivial step in this method is the computation of eigenvalues. However, these can be numerically computed by using, e.g., the QR algorithm [11][12][28] (for an explanatory treatement, see, e.g., [15]).

However, for not extremely large values of n, one could consider using a simple (but sufficiently efficient) algorithm with time complexity $O(\log n)$ (m is considered to be negligible compared to n). Anyhow, in this report we are not interested in numerical computation, but solely in theoretical matters of this method.

Finally, in order to practically demonstrate the above described method, we shall work out few examples. Example applications of the method to solving problems concerning deterministic finite automata can be found in Section 2.1.

[8] From the definition $\binom{n}{j} = \frac{n(n-1)\ldots(n-j+1)}{j!}$.

[9] If P is a predicate, then $[P] = 1$, if P is true, and $[P] = 0$ otherwise.

Example A.3.6 Let us consider an initial value problem for a homogeneous system of first-order linear OΔEs given by

$$\begin{aligned} x_1(n) &= 3x_1(n-1) + 2x_3(n-1) \\ x_2(n) &= -2x_1(n-1) + x_2(n-1) - 2x_3(n-1) \\ x_3(n) &= 2x_3(n-1) \end{aligned}$$

for $n \geq 1$, with initial conditions $x_1(0) = 1, x_2(0) = 1, x_3(0) = 2$. We shall solve this initial value problem successively for all unknown functions x_1, x_2, x_3 (although in practical applications of this method to problems concerning automata, we shall be interested only in determining one of the functions).

The first step of the method is to write down the system in a matrix form. If we introduce a notation $\mathbf{x}_n = (x_1(n), x_2(n), x_3(n))^T$, we shall clearly get

$$\mathbf{x}_n = \begin{pmatrix} 3 & 0 & 2 \\ -2 & 1 & -2 \\ 0 & 0 & 2 \end{pmatrix} \cdot \mathbf{x}_{n-1}, \qquad n \geq 1,$$

with initial conditions given by $\mathbf{x}_0 = (1, 1, 2)^T$.

The next step is to compute the eigenvalues of the matrix of the system. The characteristic polynomial of the matrix is

$$\mathrm{ch}(\lambda) = \begin{vmatrix} 3-\lambda & 0 & 2 \\ -2 & 1-\lambda & -2 \\ 0 & 0 & 2-\lambda \end{vmatrix} = (3-\lambda)(1-\lambda)(2-\lambda).$$

Therefore, the eigenvalues clearly are

$$\lambda_1 = 1, \quad \lambda_2 = 2, \quad \lambda_3 = 3$$

and their multiplicities are

$$\alpha_1 = \alpha_2 = \alpha_3 = 1,$$

i.e., all of the eigenvalues are simple.

Now, we shall find a closed form for x_1. According to Corollary A.3.5,

$$x_1(n) = c_{1,1,1} \cdot 1^n + c_{2,1,1} \cdot 2^n + c_{3,1,1} \cdot 3^n \tag{A.14}$$

for some constants $c_{1,1,1}, c_{2,1,1}, c_{3,1,1}$. As a final step, we shall determine these constants. From initial conditions, we have

$$\mathbf{x}_0 = (1, 1, 2)^T.$$

Moreover, we can easily compute

$$\mathbf{x}_1 = \begin{pmatrix} 3 & 0 & 2 \\ -2 & 1 & -2 \\ 0 & 0 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} = (7, -5, 4)^T,$$

and

$$\mathbf{x}_2 = \begin{pmatrix} 3 & 0 & 2 \\ -2 & 1 & -2 \\ 0 & 0 & 2 \end{pmatrix}^2 \cdot \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} = (29, -27, 8)^T.$$

From that we have $x_1(0) = 1$, $x_1(1) = 7$, and $x_1(2) = 29$. Thus, from (A.14) we obtain a system of linear equations

$$\begin{aligned} c_{1,1,1} \cdot 1^0 + c_{2,1,1} \cdot 2^0 + c_{3,1,1} \cdot 3^0 &= 1 \\ c_{1,1,1} \cdot 1^1 + c_{2,1,1} \cdot 2^1 + c_{3,1,1} \cdot 3^1 &= 7 \\ c_{1,1,1} \cdot 1^2 + c_{2,1,1} \cdot 2^2 + c_{3,1,1} \cdot 3^2 &= 29 \end{aligned}$$

with $c_{1,1,1}, c_{2,1,1}$, and $c_{3,1,1}$ as the unknowns. By solving this system by Gaussian elimination, we get

$$\begin{aligned} c_{1,1,1} &= 0, \\ c_{2,1,1} &= -4, \\ c_{3,1,1} &= 5. \end{aligned}$$

The solution for x_1 therefore is

$$x_1(n) = 5 \cdot 3^n - 4 \cdot 2^n.$$

Next, we shall find a closed form solution for x_2. According to Corollary A.3.5, the solution is the same as in the case of x_1, i.e.,

$$x_2(n) = c_{1,1,2} \cdot 1^n + c_{2,1,2} \cdot 2^n + c_{3,1,2} \cdot 3^n \tag{A.15}$$

for some constants $c_{1,1,2}, c_{2,1,2}, c_{3,1,2}$. From this we get a system of linear equations

$$\begin{aligned} c_{1,1,2} \cdot 1^0 + c_{2,1,2} \cdot 2^0 + c_{3,1,2} \cdot 3^0 &= 1 \\ c_{1,1,2} \cdot 1^1 + c_{2,1,2} \cdot 2^1 + c_{3,1,2} \cdot 3^1 &= -5 \\ c_{1,1,2} \cdot 1^2 + c_{2,1,2} \cdot 2^2 + c_{3,1,2} \cdot 3^2 &= -27 \end{aligned}$$

with $c_{1,1,2}, c_{2,1,2}$, and $c_{3,1,2}$ as the unknowns. By solving this system, we obtain the values of the coefficients

$$\begin{aligned} c_{1,1,2} &= 2, \\ c_{2,1,2} &= 4, \\ c_{3,1,2} &= -5. \end{aligned}$$

Thus,

$$x_2(n) = -5 \cdot 3^n + 4 \cdot 2^n + 2.$$

Finally, we shall find a closed form solution for the unknown function x_3. Once again, the solution has a form

$$x_3(n) = c_{1,1,3} \cdot 1^n + c_{2,1,3} \cdot 2^n + c_{3,1,3} \cdot 3^n \tag{A.16}$$

for some constants $c_{1,1,3}, c_{2,1,3}, c_{3,1,3}$. For these, we obtain a system of linear equations

$$\begin{aligned} c_{1,1,3} \cdot 1^0 + c_{2,1,3} \cdot 2^0 + c_{3,1,3} \cdot 3^0 &= 2 \\ c_{1,1,3} \cdot 1^1 + c_{2,1,3} \cdot 2^1 + c_{3,1,3} \cdot 3^1 &= 4 \\ c_{1,1,3} \cdot 1^2 + c_{2,1,3} \cdot 2^2 + c_{3,1,3} \cdot 3^2 &= 8 \end{aligned}$$

with $c_{1,1,3}, c_{2,1,3}$, and $c_{3,1,3}$ as the unknowns. By solving this system, we get the solution for the coefficients

$$\begin{aligned} c_{1,1,3} &= 0, \\ c_{2,1,3} &= 2, \\ c_{3,1,3} &= 0. \end{aligned}$$

Thus,

$$x_3(n) = 2 \cdot 2^n = 2^{n+1}.$$

Example A.3.7 This example will be rather trivial, but we would like to demonstrate that the above described method can be also applied to solve systems of this kind. Let us consider an initial value problem for a homogeneous system of first-order linear OΔEs given by

$$\begin{aligned} x_1(n) &= x_2(n-1) \\ x_2(n) &= 0 \end{aligned}$$

for $n \geq 1$, and with initial conditions given by $x_1(0) = 0, x_2(0) = 1$. We are interested in finding a closed form solution for x_1. If we denote $\mathbf{x}_n = (x_1(n), x_2(n))^T$, we can write the system in a matrix form as

$$\mathbf{x}_n = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \cdot \mathbf{x}_{n-1}, \qquad n \geq 1,$$

with initial conditions $\mathbf{x}_0 = (0,1)^T$.

The characteristic polynomial of the matrix is

$$\mathrm{ch}(\lambda) = \begin{vmatrix} -\lambda & 1 \\ 0 & -\lambda \end{vmatrix} = \lambda^2.$$

Thus, the matrix has only one distinct eigenvalue $\lambda_1 = 0$ with multiplicity $\alpha_1 = 2$. Thus, according to Corollary A.3.5, the solution for x_1 has a form

$$x_1(n) = c_{n=0,1} \cdot [n=0] + c_{n=1,1} \cdot [n=1]$$

for some constants $c_{n=0,1}$, and $c_{n=1,1}$. From the initial conditions, we have

$$\mathbf{x}_0 = (0,1)^T,$$

and

$$\mathbf{x}_1 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = (1,0)^T.$$

Thus, we can determine the unknown coefficients $c_{n=0,1}$ and $c_{n=1,1}$ from the system of linear equations

$$\begin{aligned} c_{n=0,1} \cdot [0=0] + c_{n=1,1} \cdot [0=1] = 0 \\ c_{n=0,1} \cdot [1=0] + c_{n=1,1} \cdot [1=1] = 1 \end{aligned}$$

with $c_{n=0,1}$ and $c_{n=1,1}$ as the unknowns. By solving these system, we determine the coefficients as

$$\begin{aligned} c_{n=0,1} &= 0, \\ c_{n=1,1} &= 1. \end{aligned}$$

Therefore, the solution for x_1 is

$$x_1(n) = [n=1],$$

which can be alternatively written as

$$x_1(n) = \begin{cases} 1 & \text{if } n = 1 \\ 0 & \text{otherwise} \end{cases}.$$

Example A.3.8 Finally, let us consider an initial value problem for a homogeneous system of first-order linear OΔEs given by

$$\begin{aligned} x_1(n) &= x_1(n-1) + 2x_2(n-1) + 5x_3(n-1) + 10x_4(n-1) \\ x_2(n) &= 2x_2(n-1) + x_3(n-1) \\ x_3(n) &= 2x_3(n-1) + 2x_4(n-1) \\ x_4(n) &= 2x_4(n-1) \end{aligned}$$

with initial conditions $x_1(0) = 1, x_2(0) = 1, x_3(0) = 2$, and $x_4(0) = 3$. We shall be interested in finding a closed form solution for x_1.

In the matrix form, the system becomes

$$\mathbf{x}_n = \begin{pmatrix} 1 & 2 & 5 & 10 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 2 \end{pmatrix} \cdot \mathbf{x}_{n-1}$$

with initial conditions given by $\mathbf{x}_0 = (1, 1, 2, 3)^T$.

The characteristic polynomial of the matrix is

$$\mathrm{ch}(\lambda) = \begin{vmatrix} 1-\lambda & 2 & 5 & 10 \\ 0 & 2-\lambda & 1 & 0 \\ 0 & 0 & 2-\lambda & 2 \\ 0 & 0 & 0 & 2-\lambda \end{vmatrix} = (1-\lambda)(2-\lambda)^3.$$

Thus, the matrix has two distinct eigenvalues

$$\lambda_1 = 1, \quad \lambda_2 = 2$$

with algebraic multiplicities

$$\alpha_1 = 1, \quad \alpha_2 = 3.$$

Thus, according to Corollary A.3.5, the solution for the unknown function x_1 has a form

$$c_{1,1,1} + c_{2,1,1} \cdot 2^n + c_{2,2,1} \cdot n2^n + c_{2,3,1} \cdot n^2 2^n$$

for some unknown constants $c_{1,1,1}, c_{2,1,1}, c_{2,2,1}$, and $c_{2,3,1}$. These constants can be determined as the solution to the system of linear equations

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 2 & 2 & 2 \\ 1 & 4 & 8 & 16 \\ 1 & 8 & 24 & 72 \end{pmatrix} \cdot \begin{pmatrix} c_{1,1,1} \\ c_{2,1,1} \\ c_{2,2,1} \\ c_{2,3,1} \end{pmatrix} = \begin{pmatrix} 1 \\ 43 \\ 161 \\ 477 \end{pmatrix}.$$

By solving this system, we get

$$\begin{aligned} c_{1,1,1} &= -19, & c_{2,1,1} &= 20, \\ c_{2,2,1} &= \frac{19}{2}, & c_{2,3,1} &= \frac{3}{2}. \end{aligned}$$

Thus, the closed form for x_1 is

$$x_1(n) = \left(\frac{3}{2}n^2 + \frac{19}{2}n + 20\right) \cdot 2^n - 19.$$

A.4 Nonnegative Matrices

In this section, we shall cover the basics of the theory of nonnegative matrices. Most importantly, we shall state and prove the *Perron-Frobenius theorem*, which plays a crucial role in the theory of equiloaded deterministic finite automata (Chapter 2). We shall present only several most important concepts of the theory, applied in this report. However, the theory of nonnegative matrices is a quite extensive subject on which whole monographs have been written. Classical treatments of the subject include [2] and [31]. Moreover, [39] also has a chapter on nonnegative matrices.

Definition A.4.1 Let A be an $n \times m$ matrix, for some n, m in $\mathbb{N}$. The matrix A is said to be *nonnegative*, if all of its entries are nonnegative. The matrix A is said to be *positive*, if all of its entries are positive.

Let us note that every transition matrix of a deterministic finite automaton (or every adjacency matrix of a digraph) is a nonnegative matrix (in fact, these matrices are nonnegative *integer* matrices). This is the main reason of the immense importance of nonnegative matrices in the theory of equiloaded DFA and DFAε.

A.4.1 Irreducible Matrices

Now, we shall define the concepts of *reducible* and *irreducible* matrices, following the definition of [31].

Definition A.4.2 Let A be an $n \times n$ square matrix, for some nonnegative integer n in $\mathbb{N}$. For $n \geq 2$, the matrix A is said to be *reducible* (or *decomposable*), if there exists an $n \times n$ permutation matrix P, such that

$$P \cdot A \cdot P^T = \begin{pmatrix} B & C \\ \mathbf{0} & D \end{pmatrix},$$

where B and D are square submatrices, and $\mathbf{0}$ is a zero submatrix. The matrix A is said to be *irreducible* (or *indecomposable*), if it is not reducible. For $n = 1$, the nonzero matrix is said to be irreducible by definition. However, we shall consider the 1×1 null matrix to be reducible.[10]

Let us note that since every permutation matrix is orthogonal, we can rewrite the condition from the previous definition also as

$$P \cdot A \cdot P^{-1} = \begin{pmatrix} B & C \\ \mathbf{0} & D \end{pmatrix}.$$

Following [31], we shall derive the alternative definition of irreducible matrices (Theorem A.4.8).

Lemma A.4.3 Let A be an $n \times n$ irreducible nonnegative matrix, for some n in $\mathbb{N}$, $n \geq 2$. Let $\mathbf{y}$ be a nonnegative column vector with n entries that has exactly k, $1 \leq k \leq n-1$, positive entries. Then $(\mathbf{I}_n + A) \cdot \mathbf{y}$ has more than k positive entries.

Proof. Let us denote

$$\mathbf{y} = (y_1, y_2, \ldots, y_n)^T.$$

By our assumption, exactly k numbers of $y_1, y_2, \ldots, y_n$ are positive and the rest $n-k$ numbers are zero. Moreover, let P be a permutation matrix, such that for

$$P \cdot \mathbf{y} = \mathbf{x} = (x_1, x_2, \ldots, x_n)^T,$$

the entries $x_1, \ldots, x_k$ are positive, and the entries $x_{k+1}, \ldots, x_n$ are zero.

Since A is nonnegative, the column vector

$$(\mathbf{I}_n + A) \cdot \mathbf{y} = \mathbf{y} + A \cdot \mathbf{y}$$

must have at least k positive entries and at most $n-k$ zero entries. We shall prove, that it has *more than* k positive entries and *less than* $n-k$ zero entries.

For the purpose of contradiction, let us suppose that it has exactly $n-k$ zero entries. This implies that if $y_i = 0$, then $(A \cdot \mathbf{y})_i = 0$, for $i = 1, \ldots, n$. Therefore $(P \cdot A \cdot \mathbf{y})_i = 0$, whenever

[10] Since otherwise, many presented theorems on irreducible matrices would have to exclude this special case. However, in [31], the 1×1 zero matrix is considered to be irreducible.

$x_i = 0$, i.e., $(P \cdot A \cdot \mathbf{y})_i = 0$ for $i = k+1, \ldots, n$. Furthermore, $\mathbf{y} = P^{-1} \cdot \mathbf{x} = P^T \cdot \mathbf{x}$ (since P is a permutation matrix, that is always orthogonal). So $(P \cdot A \cdot P^T \cdot \mathbf{x})_i = 0$ for $i = k+1, \ldots, n$.

Now, let us denote $R = P \cdot A \cdot P^T = (r_{i,j})_{n \times n}$. Then

$$(R \cdot \mathbf{x})_i = \sum_{j=1}^{n} r_{i,j} x_j = \sum_{j=1}^{k} r_{i,j} x_j = 0$$

for $i = k+1, \ldots, n$. But since $x_j > 0$ for $j = 1, \ldots, k$, we have $r_{i,j} = 0$ for $i = k+1, \ldots, n$ and $j = 1, \ldots, k$. Thus the matrix $R = P \cdot A \cdot P^T$ has a form

$$R = P \cdot A \cdot P^T = \begin{pmatrix} B & C \\ \mathbf{0} & D \end{pmatrix},$$

where B and D are square submatrices, and $\mathbf{0}$ is a zero submatrix. Thus, A is reducible, and that contradicts our assumption. □

Corollary A.4.4 Let A be an $n \times n$ irreducible nonnegative matrix, for some n in $\mathbb{N}$, $n \geq 1$. Let $\mathbf{y}$ be a nonzero nonnegative column vector with n entries. Then $(\mathbf{I}_n + A)^{n-1} \cdot \mathbf{y}$ is a positive column vector.

Proof. According to Lemma A.4.3, for $n \geq 2$, $(\mathbf{I}_n + A)^i \cdot \mathbf{y}$ has at least one more positive entry than $(\mathbf{I}_n + A)^{i-1} \cdot \mathbf{y}$, for $i = 1, \ldots, n-1$. Since $\mathbf{y} = (\mathbf{I}_n + A)^0 \cdot \mathbf{y}$ has at least one positive entry, $(\mathbf{I}_n + A)^{n-1} \cdot \mathbf{y}$ must have at least n positive entries, i.e., $(\mathbf{I}_n + A)^{n-1} \cdot \mathbf{y}$ is a positive column vector. For $n = 1$, the statement of the corollary is obvious. □

Corollary A.4.5 Let A be an $n \times n$ nonnegative matrix, for some n in $\mathbb{N}$, $n \geq 1$. The matrix A is irreducible if and only if $(\mathbf{I}_n + A)^{n-1}$ is a positive matrix.

Proof. Let the matrix A be irreducible. Let us denote the unit column vectors with n entries by

$$\mathbf{e}_i = (\underbrace{0, \ldots, 0}_{i-1}, 1, \underbrace{0, \ldots, 0}_{n-i})^T, \qquad i = 1, \ldots, n.$$

Then, according to Corollary A.4.4, $(\mathbf{I}_n + A)^{n-1} \cdot \mathbf{e}_i$ is a positive vector for $i = 1, \ldots, n$. In other words, all columns of the matrix $(\mathbf{I}_n + A)^{n-1}$ are positive. Thus, $(\mathbf{I}_n + A)^{n-1}$ is a positive matrix.

On the contrary, let $(\mathbf{I}_n + A)^{n-1}$ be a positive matrix. It is clear that each positive matrix is irreducible, and so $(\mathbf{I}_n + A)^{n-1}$ is an irreducible matrix. But this implies that $\mathbf{I}_n + A$ is also irreducible, since if it was reducible, we could write

$$P \cdot (\mathbf{I}_n + A) \cdot P^{-1} = \begin{pmatrix} B & C \\ \mathbf{0} & D \end{pmatrix}$$

and therefore

$$P \cdot (\mathbf{I}_n + A)^{n-1} \cdot P^{-1} = \left(P \cdot \mathbf{I}_n + A \cdot P^{-1}\right)^{n-1} = \begin{pmatrix} B & C \\ \mathbf{0} & D \end{pmatrix}^{n-1} = \begin{pmatrix} B' & C' \\ \mathbf{0} & D' \end{pmatrix},$$

i.e., $(\mathbf{I}_n + A)^{n-1}$ would be reducible. That is, $\mathbf{I}_n + A$ is irreducible. But then also A is irreducible, since for each matrix M, the diagonal entries of M have effect only on diagonal entries of the matrix $P \cdot M \cdot P^T$, but no effect on off-diagonal entries. The matrix A is therefore irreducible and the corollary is proven. □

Lemma A.4.6 Let A be an $n \times n$ irreducible nonnegative matrix, for some n in $\mathbb{N}$. Then A does not have any zero column.

Proof. For the purpose of contradiction, let us suppose that the s-th column of the matrix A is zero, for some s in $\{1, \ldots, n\}$. Let us define the permutation matrix P by

$$P = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{pmatrix},$$

where

$$p_{i,j} = \begin{cases} 1 & \text{if } (i,j) = (1,s) \text{ or } (i,j) = (s,1) \text{ or } 1 \neq i = j \neq s \\ 0 & \text{otherwise} \end{cases}, \qquad i,j = 1, \ldots, n.$$

The left-multiplication of A with P does only permute the rows of the matrix A, and the zero column is still present in the matrix $P \cdot A$. The right-multiplication with P^T permutes columns, and therefore the first column of the matrix $P \cdot A \cdot P^T$ is zero. Now, if we define the submatrix B to be the 1×1 null matrix, it is clear that the matrix $P \cdot A \cdot P^T$ has a form

$$P \cdot A \cdot P^T = \begin{pmatrix} B & C \\ \mathbf{0} & D \end{pmatrix},$$

where B and D are square submatrices, and $\mathbf{0}$ is a zero submatrix. Thus, the matrix A is reducible, i.e., a contradiction. $\square$

Lemma A.4.7 Let A be an $n \times n$ irreducible nonnegative matrix, for some n in $\mathbb{N}$. Then A does not have any zero row.

Proof. The lemma can be easily proved in a similar manner as Lemma A.4.6. $\square$

Theorem A.4.8 Let A be a nonnegative $n \times n$ matrix for some n in $\mathbb{N}$,

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix}.$$

For k in $\mathbb{N}$, let us denote[11]

$$A^k = \begin{pmatrix} a^{(k)}_{1,1} & a^{(k)}_{1,2} & \cdots & a^{(k)}_{1,n} \\ a^{(k)}_{2,1} & a^{(k)}_{2,2} & \cdots & a^{(k)}_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a^{(k)}_{n,1} & a^{(k)}_{n,2} & \cdots & a^{(k)}_{n,n} \end{pmatrix}.$$

The matrix A is irreducible if and only if for all i, j in $\{1, \ldots, n\}$, nonnegative integer k in $\mathbb{N}$ exists, such that $a^{(k)}_{i,j} > 0$.

Proof. Let A be irreducible. From the Corollary A.4.5 it follows, that the matrix $(\mathbf{I}_n + A)^{n-1}$ is positive. Let us denote

$$A' = (a'_{i,j})_{n \times n} = (\mathbf{I}_n + A)^{n-1} \cdot A.$$

Since $(\mathbf{I}_n + A)^{n-1}$ is positive and, according to Lemma A.4.6, A does not have any zero column, the matrix A' is positive as well. Let $c_1, c_2, \ldots, c_{n-1}$ in $\mathbb{R}$ be constants, such that

$$A' = A^n + c_{n-1} \cdot A^{n-1} + \ldots + c_2 \cdot A^2 + c_1 \cdot A,$$

[11] We shall use this notation also later on in this section.

i.e.,

$$a'_{i,j} = a^{(n)}_{i,j} + c_{n-1} \cdot a^{(n-1)}_{i,j} + \ldots + c_2 \cdot a^{(2)}_{i,j} + c_1 \cdot a^{(1)}_{i,j}$$

for all i, j in $\{1, \ldots, n\}$. But since $a'_{i,j} > 0$ for all i, j in $\{1, \ldots, n\}$, it follows that for all i, j in $\{1, \ldots, n\}$ there must be a nonnegative integer k in $\{1, \ldots, n\}$ such that $a^{(k)}_{i,j} > 0$.

Now, let us prove the converse statement. We shall prove, that if the matrix A is reducible, then there exist i, j in $\{1, \ldots, n\}$, such that $a^{(k)}_{i,j} = 0$ for all k in $\mathbb{N}$. Since A is reducible, a permutation matrix P exists, such that

$$P \cdot A \cdot P^T = \begin{pmatrix} B & C \\ \mathbf{0} & D \end{pmatrix},$$

where B and D are square submatrices, and $\mathbf{0}$ is a zero submatrix. Let B be an $s \times s$ submatrix. Clearly, for all (i, j) with i in $\{s+1, \ldots, n\}$ and with j in $\{1, \ldots, s\}$, the (i, j) entry of $P \cdot A^k \cdot P^T$ is zero for all k in $\mathbb{N}$. But there clearly is a one-to-one correspondence between entries of $P \cdot M \cdot P^T$ and entries of M, that is the same for all matrices M of size $n \times n$ (for fixed P). Thus, the matrix A also has the entry (i', j') in $\{1, \ldots, n\}^2$, such that $a^{(k)}_{i',j'} = 0$ for all k in $\mathbb{N}$. The theorem is therefore proven. $\square$

As already mentioned, Theorem A.4.8 provides the alternative definition of irreducible matrices. In fact, the manipulation with the definition according to Theorem A.4.8 is sometimes more easier than the manipulation with the original formulation from Definition A.4.2 (at least for the purposes of this report). In some texts, the definition provided by Theorem A.4.8 is actually used as a primary definition (e.g., in [5]). However, the definition provided by Theorem A.4.8 says almost nothing about the etymology of the term *irreducible*.

There is an interesting connection of the notion of an irreducible matrix to graph theory (and as a consequence also to the theory of deterministic finite automata). From Theorem A.4.8, it is clear that a digraph has an irreducible adjacency matrix if and only if the digraph is strongly connected and different from the isolated vertex without any loops (because the entry (i, j) of the k-th power of the adjacency matrix is exactly the number of walks of length k from the i-th vertex to the j-th vertex). In other words, a deterministic finite automaton has an irreducible transition matrix if and only if its graphical representation is strongly connected and different from the isolated state without any transitions.

However, this can also be seen, after a little effort, directly from Definition A.4.2. A transformation $P \cdot A \cdot P^T$ of an adjacency matrix A of a given graph, where P is a permutation matrix, can be interpreted simply as a relabeling of vertices. The graph therefore has a reducible transition matrix if and only if its set of vertices $V(G)$ can be decomposed into two sets V_1 and V_2 (corresponding to submatrices B and D in Definition A.4.2) such that there is not any edge from vertex in V_2 to vertex in V_1. If we would contract the vertices of both of this sets into a single vertex, the resulting graph would be clearly a directed acyclic graph. In other words, the graph is not strongly connected. The converse can be observed in a similar manner. The role of permutation matrix P is only to relabel the vertices in order to "bring together" the vertices in the same strongly connected component.

In the following theorem, we shall show that every nonnegative eigenvector of an irreducible nonnegative matrix is strictly positive.

Theorem A.4.9 Let A be an $n \times n$ nonnegative irreducible matrix. Let $\mathbf{x}$ be a nonnegative eigenvector of A with corresponding eigenvalue λ. Then $\mathbf{x}$ is positive.

Proof. Since $\mathbf{x}$ is an eigenvector with corresponding eigenvalue λ,

$$A \cdot \mathbf{x} = \lambda \cdot \mathbf{x}. \tag{A.17}$$

Since both the matrix A and the eigenvector $\mathbf{x}$ are nonnegative, (A.17) implies that λ has to be nonnegative as well. Moreover, from (A.17) we have

$$(\mathbf{I}_n + A) \cdot \mathbf{x} = (1 + \lambda) \cdot \mathbf{x}. \tag{A.18}$$

For the purpose of contradiction, let us suppose that $\mathbf{x}$ is not positive, i.e., that $\mathbf{x}$ has $k > 0$ zero entries. It is clear that $(1+\lambda) \cdot \mathbf{x}$ has k zero entries as well. However, according to Lemma A.4.3, $(\mathbf{I}_n + A) \cdot \mathbf{x}$ has less than k zero entries, which contradicts (A.18). Thus, $\mathbf{x}$ cannot have a zero entry, i.e., the eigenvector $\mathbf{x}$ is strictly positive. □

A.4.2 The Collatz-Wielandt Function

In this subsection, we shall define and study some properties of the Collatz-Wielandt function. The Collatz-Wielandt function will be of crucial importance for the proof of the Perron-Frobenius theorem, presented in Subsection A.4.5. In this subsection, we shall follow [31]. Moreover, in this subsection we shall use the symbol $\mathbb{P}$ to denote the set of nonnegative real numbers.

Definition A.4.10 (The Collatz-Wielandt Function [7] [40]) Let A be an $n \times n$ irreducible nonnegative matrix, for some n in $\mathbb{N}$. *The Collatz-Wielandt function* associated with the matrix A is the function $f_A : \mathbb{P}^n \setminus \{\mathbf{0}\} \to \mathbb{P}$ defined by

$$f_A(\mathbf{x}) = \min_{x_i \neq 0} \frac{\left(A \cdot \mathbf{x}^T\right)_i}{x_i}$$

for all nonzero $\mathbf{x} = (x_1, \ldots, x_n)$ in $\mathbb{P}^n$.[12]

In the following lemma, we shall prove several basic properties of this function. As a consequence of the lemma, we shall also acquire an intuitive understanding of the concept of the Collatz-Wielandt function.

Lemma A.4.11 Let A be an $n \times n$ irreducible nonnegative matrix, for some nonnegative integer n in $\mathbb{N}$. Let $f_A : \mathbb{P}^n \setminus \{\mathbf{0}\} \to \mathbb{P}$ be the Collatz-Wielandt function associated with A. Then the following three properties hold:

1. The function f_A is homogeneous of degree zero, i.e., for all $\mathbf{x}$ in $\mathbb{P}^n \setminus \{\mathbf{0}\}$ and for all α in $\mathbb{P} \setminus \{0\}$,
$$f_A(\alpha \cdot \mathbf{x}) = \alpha^0 \cdot f_A(\mathbf{x}) = f_A(\mathbf{x}).$$

2. For given $\mathbf{x}$ in $\mathbb{P}^n \setminus \{\mathbf{0}\}$, $f_A(\mathbf{x})$ can be defined by
$$f_A(\mathbf{x}) = \max\left\{\rho \in \mathbb{P} \;\middle|\; A \cdot \mathbf{x}^T \geq \rho \cdot \mathbf{x}^T\right\},$$
i.e., $f_A(\mathbf{x})$ is the largest nonnegative[13] real number ρ, such that $A \cdot \mathbf{x}^T \geq \rho \cdot \mathbf{x}^T$.

3. For given $\mathbf{x}$ in $\mathbb{P}^n \setminus \{\mathbf{0}\}$ and $\mathbf{y} = \left((\mathbf{I}_n + A)^{n-1} \cdot \mathbf{x}^T\right)^T$, the inequality $f_A(\mathbf{y}) \geq f_A(\mathbf{x})$ holds.

Proof. We shall prove each claim separately.

1. Let $\mathbf{x} = (x_1, \ldots, x_n)$ be in $\mathbb{P}^n \setminus \{\mathbf{0}\}$ and α be in $\mathbb{P} \setminus \{0\}$. Then,
$$f_A(\alpha \cdot \mathbf{x}) = \min_{x_i \neq 0} \frac{\left(A \cdot \left(\alpha \cdot \mathbf{x}^T\right)\right)_i}{\alpha \cdot x_i} = \min_{x_i \neq 0} \frac{\alpha \cdot \left(A \cdot \mathbf{x}^T\right)_i}{\alpha \cdot x_i} = \min_{x_i \neq 0} \frac{\left(A \cdot \mathbf{x}^T\right)_i}{x_i} = f_A(\mathbf{x}).$$

[12] In the context of the Collatz-Wielandt function (unlike in the rest of this report) we work primarily with row vectors instead of column vectors. The reason for this is notational convenience (despite some complications with using transposes).

[13] It is obvious that $A \cdot \mathbf{x}^T \geq 0$, so the number ρ is well defined. The number ρ therefore could be, equivalently, defined to be the largest such *real* number.

2. It follows directly from the definition of the Collatz-Wielandt function that

$$A \cdot \mathbf{x}^T \geq f_A(\mathbf{x}) \cdot \mathbf{x}^T$$

and that for at least one coordinate k in $\{1, \ldots, n\}$, $\left(A \cdot \mathbf{x}^T\right)_k = \left(f_A(\mathbf{x}) \cdot \mathbf{x}^T\right)_k$. Thus, for all $\rho' > f_A(\mathbf{x})$, there is at least one coordinate k in $\{1, \ldots, n\}$, such that $\left(\rho' \cdot \mathbf{x}^T\right)_k$ is greater than $\left(A \cdot \mathbf{x}^T\right)_k$. The claim is therefore proved.

3. As we have observed,

$$A \cdot \mathbf{x}^T \geq f_A(\mathbf{x}) \cdot \mathbf{x}^T. \tag{A.19}$$

By multiplying both sides of (A.19) by $(\mathbf{I}_n + A)^{n-1}$, we obtain

$$(\mathbf{I}_n + A)^{n-1} \cdot A \cdot \mathbf{x}^T \geq f_A(\mathbf{x}) \cdot (\mathbf{I}_n + A)^{n-1} \cdot \mathbf{x}^T. \tag{A.20}$$

Now, clearly $A \cdot (\mathbf{I}_n + A)^{n-1} = (\mathbf{I}_n + A)^{n-1} \cdot A$, that is, we can rewrite (A.20) as

$$A \cdot (\mathbf{I}_n + A)^{n-1} \cdot \mathbf{x}^T \geq f_A(\mathbf{x}) \cdot (\mathbf{I}_n + A)^{n-1} \cdot \mathbf{x}^T,$$

i.e., $A \cdot \mathbf{y}^T \geq f_A(\mathbf{x}) \cdot \mathbf{y}^T$. The result $f_A(\mathbf{y}) \geq f_A(\mathbf{x})$ now follows from the claim 2.

The lemma is proved. □

As we have already anticipated, it is possible to acquire an intuitive understanding of the concept of the Collatz-Wielandt function from Lemma A.4.11. From the claim 2 of the lemma, it is clear that if $\mathbf{x}^T$ is the (right) eigenvector of the matrix A, then $f_A(\mathbf{x})$ is equal to the eigenvalue corresponding to $\mathbf{x}^T$ (since both the matrix A and the eigenvector $\mathbf{x}^T$ are real and nonnegative, the eigenvalue corresponding to $\mathbf{x}^T$ has to be real and nonnegative as well). If $\mathbf{x}^T$ is not the eigenvector, then the value of the Collatz-Wielandt function at $\mathbf{x}$ can be interpreted as a "lower eigenvalue" corresponding to $\mathbf{x}^T$. In other words, the matrix A "behaves as" the scalar value $f_A(\mathbf{x})$ for at least one coordinate of $\mathbf{x}^T$, and for each other coordinate of $\mathbf{x}^T$, the matrix A "behaves as" some larger scalar value.

Notation A.4.12 Let n be in $\mathbb{N}$. By E^n, we denote the *standard* $(n-1)$*-simplex*, i.e., the set

$$E^n = \left\{ (x_1, \ldots, x_n) \in \mathbb{P}^n \;\middle|\; \sum_{i=1}^{n} x_i = 1 \right\}.$$

The following theorem is of crucial importance for the proof of the Perron-Frobenius theorem, presented in Subsection A.4.5.

Theorem A.4.13 Let A be an $n \times n$ irreducible nonnegative matrix, for some n in $\mathbb{N}$. Then the Collatz-Wielandt function f_A associated with A attains a maximum in E^n.

Proof. Let us define the set G by

$$G = \left\{ \left((\mathbf{I}_n + A)^{n-1} \cdot \mathbf{x}^T \right)^T \;\middle|\; \mathbf{x} \in E^n \right\}.$$

The set G can be clearly seen to be a closed and bounded subset of $\mathbb{R}^n$, and thus, G is compact.[14] By Corollary A.4.5, $(\mathbf{I}_n + A)^{n-1}$ is a positive matrix, and therefore all n-tuples in G have all components strictly positive. It can be easily seen that f_A is continuous at any such n-tuple, thus f_A is continuous on G.

[14] This follows from the *Generalized Heine-Borel theorem* (see, e.g., [35]).

Every continuous function defined on a compact subset S of $\mathbb{R}^n$ attains a maximum in S.[15] Thus, f_A attains a maximum in G (at some point $\mathbf{y}^{max} = (y_1^{max}, \ldots, y_n^{max})$ in G). Now, let $\mathbf{x}_{max}$ be defined by

$$\mathbf{x}^{max} = (x_1^{max}, \ldots, x_n^{max}) = \frac{1}{\sum_{i=1}^n y_i^{max}} \cdot \mathbf{y}^{max}.$$

Clearly, $\sum_{i=1}^n x_i^{max} = 1$, and thus $\mathbf{x}^{max}$ is in E^n. Now, let $\mathbf{x}$ be an arbitrary vector in E^n, and let $\mathbf{y}$ in G be defined by $\mathbf{y} = \left((\mathbf{I}_n + A)^{n-1} \cdot \mathbf{x}^T\right)^T$. By the claim 3 of Lemma A.4.11, we have $f_A(\mathbf{x}) \leq f_A(\mathbf{y})$. By the maximality of $\mathbf{y}^{max}$ in G, we have $f_A(\mathbf{y}) \leq f_A(y^{max})$. And finally, by the claim 1 of Lemma A.4.11, we obtain

$$f_A(y^{max}) = f_A\left(\left(\sum_{i=1}^n y_i^{max}\right) \cdot \mathbf{x}^{max}\right) = f_A(\mathbf{x}^{max}).$$

Thus, $f_A(\mathbf{x}) \leq f_A(\mathbf{x}^{max})$ for all $\mathbf{x}$ in E^n, i.e., the function f_A attains a maximum in E^n at the point $\mathbf{x}^{max}$. □

A.4.3 The Perron-Frobenius Eigenvalue

In this subsection, we shall in fact prove a part of the Perron-Frobenius theorem and show that every irreducible nonnegative matrix has a unique (as to the algebraic multiplicity) real eigenvalue equal to its spectral radius, referred to as the *Perron-Frobenius eigenvalue*. However, the Perron-Frobenius theorem involves more claims, and we shall state and prove the complete theorem in Subsection A.4.5. We shall follow the presentation of [31].

Theorem A.4.14 Let A be an $n \times n$ irreducible nonnegative matrix,

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix},$$

for some n in $\mathbb{N}$. Let ρ be the spectral radius of A. Then ρ is an eigenvalue of A, referred to as the *Perron-Frobenius eigenvalue* [4] of the matrix A.[16] Moreover, there is a positive eigenvector $\mathbf{x}_\rho$ in E^n corresponding to ρ[17] and the formulas[18]

$$\rho = \max_{\mathbf{x} \in E^n} f_A(\mathbf{x}), \qquad \mathbf{x}_\rho = \left(\arg\max_{\mathbf{x} \in E^n} f_A(\mathbf{x})\right)^T$$

hold.

Proof. According to Theorem A.4.13, a vector $\mathbf{x}^{max}$ in E^n exists, such that $f_A(\mathbf{x}^{max}) \geq f_A(\mathbf{x})$ for all $\mathbf{x}$ in E^n. Let us define

$$\lambda := f_A(\mathbf{x}^{max}).$$

The number λ is clearly positive, since for $\mathbf{u} = \frac{1}{n} \cdot (\underbrace{1, 1, \ldots, 1}_{n})$ we have

$$\lambda \geq f_A(\mathbf{u}) = \min_{i=1,\ldots,n} \frac{\left(A \cdot \mathbf{u}^T\right)_i}{\frac{1}{n}} = \min_{i=1,\ldots,n} \sum_{j=1}^n a_{i,j} > 0,$$

[15] See, e.g., [29].
[16] The eigenvalue ρ is also referred to as the *maximal eigenvalue* [31], or as the *leading eigenvalue* [18] of the matrix A.
[17] Referred to as the *Perron-Frobenius eigenvector*, the *maximal eigenvector*, or as the *leading eigenvector*.
[18] The first formula is referred to as the *Collatz-Wielandt formula* [30].

where the last inequality follows from the fact that an irreducible matrix cannot have a zero row (Lemma A.4.7). We shall first show that λ is an eigenvalue, and that there is a positive eigenvector corresponding to λ. Later, we shall show that λ is equal to the spectral radius ρ.

Let us show that λ is an eigenvalue of A with corresponding eigenvector $(\mathbf{x}^{max})^T$. According to the claim 2 of Lemma A.4.11, $A \cdot (\mathbf{x}^{max})^T \geq \lambda \cdot (\mathbf{x}^{max})^T$. We shall show that $A \cdot (\mathbf{x}^{max})^T = \lambda \cdot (\mathbf{x}^{max})^T$, i.e., that λ is an eigenvalue and that $(\mathbf{x}^{max})^T$ is a corresponding eigenvector. For the purpose of contradiction, let us suppose that $A \cdot (\mathbf{x}^{max})^T \neq \lambda \cdot (\mathbf{x}^{max})^T$. Then

$$A \cdot (\mathbf{x}^{max})^T - \lambda \cdot (\mathbf{x}^{max})^T$$

is a nonzero nonnegative column vector. Thus, according to Corollary A.4.4, the column vector

$$(\mathbf{I}_n + A)^{n-1} \cdot \left(A \cdot (\mathbf{x}^{max})^T - \lambda \cdot (\mathbf{x}^{max})^T \right)$$

is positive. Now, let us define $\mathbf{y}^{max} = (y_1^{max}, \ldots, y_n^{max}) = \left((\mathbf{I}_n + A)^{n-1} \cdot (\mathbf{x}^{max})^T \right)^T$. Since clearly

$$(\mathbf{I}_n + A)^{n-1} \cdot A = A \cdot (\mathbf{I}_n + A)^{n-1},$$

the column vector

$$A \cdot (\mathbf{y}^{max})^T - \lambda \cdot (\mathbf{y}^{max})^T$$

is positive as well. Thus, a positive number ε exists, such that the column vector

$$A \cdot (\mathbf{y}^{max})^T - (\lambda + \varepsilon) \cdot (\mathbf{y}^{max})^T$$

is nonnegative. Thus, by the claim 2 of Lemma A.4.11,

$$\lambda + \varepsilon \leq f_A\left(\mathbf{y}^{max}\right),$$

i.e.,

$$\lambda < f_A\left(\mathbf{y}^{max}\right). \tag{A.21}$$

Now, let us define $\mathbf{x}^<$ by

$$\mathbf{x}^< = \frac{1}{\sum_{i=1}^n y_i^{max}} \cdot \mathbf{y}^{max}.$$

Then by the claim 1 of Lemma A.4.11,

$$f_A\left(\mathbf{y}^{max}\right) = f_A\left(\mathbf{x}^<\right).$$

Thus, (A.21) implies

$$\lambda < f_A\left(\mathbf{x}^<\right),$$

which, since $\mathbf{x}^<$ is clearly in E^n, contradicts the definition of λ. Thus, λ is an eigenvalue and $\mathbf{x}^{max}$ is a corresponding eigenvector. Since $\mathbf{x}^{max}$ is in E^n, $\mathbf{x}^{max}$ is nonnegative, and therefore, by Theorem A.4.9, positive.

It remains to show that $\lambda = \rho$. Since λ is a positive real number, it suffices to show that for all eigenvalues λ' of A, $|\lambda'| \leq \lambda$.

In fact, let λ' in $\mathbb{C}$ be an eigenvalue of A and $\mathbf{z} = (z_1, \ldots, z_n)^T$ be a corresponding eigenvector, i.e.,

$$A \cdot \mathbf{z} = \lambda' \cdot \mathbf{z}.$$

Then,

$$\lambda' \cdot z_t = \sum_{j=1}^n a_{t,j} \cdot z_j, \qquad t = 1, \ldots, n,$$

and thus

$$\left|\lambda'\right| \cdot |z_t| \leq \sum_{j=1}^{n} a_{t,j} \cdot |z_j|, \qquad t = 1, \ldots, n.$$

That is,

$$\left|\lambda'\right| \cdot |\mathbf{z}| \leq A \cdot |\mathbf{z}|,$$

where $|\mathbf{z}|$ denotes the column vector $|\mathbf{z}| = (|z_1|, \ldots, |z_n|)^T$. Thus,

$$\left|\lambda'\right| \leq f_A(|\mathbf{z}|) = f_A\left(\frac{1}{\sum_{i=1}^{n} |z_i|} \cdot |\mathbf{z}|\right) \leq \lambda,$$

where the first step is by the claim 2 of Lemma A.4.11, the second step is by the claim 1 of Lemma A.4.11 (the vector $|\mathbf{z}|$ is being normed as to belong to E^n), and the third step is by the definition of λ. □

In the following lemma, we shall observe that the dimension of the eigenspace corresponding to the Perron-Frobenius eigenvalue (i.e., its geometric multiplicity) is always 1. We shall use the lemma in the proof of Theorem A.4.16, where we shall show the same for the algebraic multiplicity of the Perron-Frobenius eigenvalue.

Lemma A.4.15 Let A be an $n \times n$ irreducible nonnegative matrix, for some n in $\mathbb{N}$. Let ρ be the Perron-Frobenius eigenvalue of the matrix A. Then the geometric multiplicity of ρ is 1.

Proof. According to Theorem A.4.14, a positive eigenvector $\mathbf{x} = (x_1, \ldots, x_n)^T$ corresponding to ρ exists. Let $\mathbf{y}$ be an arbitrary eigenvector of A corresponding to ρ. We shall show that $\mathbf{x}$ and $\mathbf{y}$ are linearly dependent and the result will follow.

Since $\mathbf{y}$ is an eigenvector corresponding to ρ, $\mathbf{y} = (y_1, \ldots, y_n)^T$ is a nonzero column vector and the identity

$$A \cdot \mathbf{y} = \rho \cdot \mathbf{y} \tag{A.22}$$

holds. From (A.22), we may derive (in exactly the same way as in the proof of Theorem A.4.14, with the use of the fact that ρ is positive) the inequality

$$A \cdot |\mathbf{y}| \geq \rho \cdot |\mathbf{y}|,$$

where $|\mathbf{y}| = (|y_1|, \ldots, |y_n|)^T$. If $A \cdot |\mathbf{y}| \neq \rho \cdot |\mathbf{y}|$, then, in the same way as in the proof of Theorem A.4.14, it is possible to conclude that for $\mathbf{y}' = (\mathbf{I}_n + A)^{n-1} \cdot |\mathbf{y}|$, the inequality

$$\rho < f_A(\mathbf{y}')$$

holds. This leads to the same contradiction as in the proof of Theorem A.4.14. Thus, $|\mathbf{y}|$ is an eigenvector of A corresponding to ρ. The eigenvector $|\mathbf{y}|$ is clearly nonnegative, and therefore, by Theorem A.4.9, positive. Thus, $\mathbf{y}$ does not have any zero entry. Specially, $y_1 \neq 0$. Moreover, this is true for all eigenvectors $\mathbf{y}$ corresponding to ρ, so we may conclude that if a column vector has a zero entry, then it is not an eigenvector corresponding to ρ.

Let us now consider a column vector $y_1 \cdot \mathbf{x} - x_1 \cdot \mathbf{y}$. The first entry of this column vector is zero, and thus it is not an eigenvector corresponding to ρ. However, it is a linear combination of eigenvectors corresponding to ρ and therefore is in the eigenspace corresponding to ρ. Thus,

$$y_1 \cdot \mathbf{x} - x_1 \cdot \mathbf{y} = \mathbf{0},$$

where $x_1, y_1 \neq 0$, i.e., the vectors $\mathbf{x}$ and $\mathbf{y}$ are linearly dependent. □

In the following theorem, we shall show that the Perron-Frobenius eigenvalue is always simple, i.e., its algebraic multiplicity is always equal to 1.

Theorem A.4.16 Let A be an $n \times n$ irreducible nonnegative matrix,

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix},$$

for some n in $\mathbb{N}$. Let ρ be the Perron-Frobenius eigenvalue of the matrix A. Then the algebraic multiplicity of ρ is 1.

Proof. Let $\mathrm{ch}(\lambda) = |A - \lambda \cdot \mathbf{I}_n|$ be the characteristic polynomial of the matrix A. We shall show that ρ is a simple root of the polynomial $\mathrm{ch}(\lambda)$. We have already proved that ρ is an eigenvalue of A, and therefore is a root of $\mathrm{ch}(\lambda)$. Thus, it remains to show that ρ is not a multiple root of $\mathrm{ch}(\lambda)$. To prove this, it suffices to show that $\mathrm{ch}'(\rho) \neq 0$, where $\mathrm{ch}'(\lambda)$ is the first derivative of $\mathrm{ch}(\lambda)$.

Let us first derive a formula for the first derivative of the determinant of a matrix of differentiable functions. Let $X(\lambda)$ be a matrix

$$X(\lambda) = \begin{pmatrix} x_{1,1}(\lambda) & x_{1,2}(\lambda) & \dots & x_{1,n}(\lambda) \\ x_{2,1}(\lambda) & x_{2,2}(\lambda) & \dots & x_{2,n}(\lambda) \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1}(\lambda) & x_{n,2}(\lambda) & \dots & x_{n,n}(\lambda) \end{pmatrix},$$

with $x_{i,j} : \mathbb{R} \to \mathbb{R}$ being a differentiable function of variable λ, for $i, j = 1, \dots, n$. Then,

$$\begin{aligned}
\frac{d}{d\lambda}|X(\lambda)| &= \frac{d}{d\lambda} \sum_{\sigma \in S_n} \mathrm{sgn}(\sigma) \cdot \prod_{i=1}^{n} x_{i,\sigma(i)}(\lambda) = \sum_{\sigma \in S_n} \mathrm{sgn}(\sigma) \cdot \left(\frac{d}{d\lambda} \prod_{i=1}^{n} x_{i,\sigma(i)}(\lambda) \right) = \\
&= \sum_{\sigma \in S_n} \mathrm{sgn}(\sigma) \cdot \sum_{i=1}^{n} \left(\frac{d}{d\lambda} x_{i,\sigma(i)}(\lambda) \right) \cdot \prod_{\substack{1 \le k \le n \\ i \ne k}} x_{k,\sigma(k)}(\lambda) = \\
&= \sum_{i=1}^{n} \sum_{\sigma \in S_n} \left(\frac{d}{d\lambda} x_{i,\sigma(i)}(\lambda) \right) \cdot \mathrm{sgn}(\sigma) \cdot \prod_{\substack{1 \le k \le n \\ i \ne k}} x_{k,\sigma(k)}(\lambda) = \\
&= \sum_{i,j=1}^{n} \left(\frac{d}{d\lambda} x_{i,j}(\lambda) \right) \cdot \sum_{\sigma' \in S_{n-1}} (-1)^{i+j} \cdot \mathrm{sgn}(\sigma') \cdot \prod_{k=1}^{n-1} x_{k+[k \ge i], \sigma'(k+[k \ge i]) + [\sigma'(k+[k \ge i]) \ge j]}(\lambda) = \\
&= \sum_{i,j=1}^{n} \left(\frac{d}{d\lambda} x_{i,j}(\lambda) \right) \cdot (-1)^{i+j} \cdot \sum_{\sigma' \in S_{n-1}} \mathrm{sgn}(\sigma') \cdot \prod_{k=1}^{n-1} x_{k+[k \ge i], \sigma'(k+[k \ge i]) + [\sigma'(k+[k \ge i]) \ge j]}(\lambda) = \\
&= \sum_{i,j=1}^{n} \left(\frac{d}{d\lambda} x_{i,j}(\lambda) \right) \cdot \mathrm{X}_{i,j}(\lambda),
\end{aligned}$$

where $X_{i,j}(\lambda)$ is the cofactor of the entry (i,j) of the matrix $X(\lambda)$, i.e., $X_{i,j}(\lambda) = (-1)^{i+j} \cdot |Y_{i,j}(\lambda)|$, where $Y_{i,j}(\lambda)$ is an $(n-1) \times (n-1)$ matrix obtained from matrix $X(\lambda)$ by deleting the i-th row and the j-th column.

Thus, we may express the derivative $\mathrm{ch}'(\lambda)$ of the characteristic polynomial (we shall denote the cofactor of the entry (i,j) of the matrix $A - \lambda \cdot \mathbf{I}_n$ by $\mathrm{cof}_{i,j}(A - \lambda \cdot \mathbf{I}_n)$) as

$$\begin{aligned}
\mathrm{ch}'(\lambda) = \frac{d}{d\lambda}|A - \lambda \cdot \mathbf{I}_n| &= \sum_{i,j=1}^{n} \left(\frac{d}{d\lambda} \left(a_{i,j} - \delta_{i,j} \cdot \lambda \right) \right) \cdot \mathrm{cof}_{i,j}(A - \lambda \cdot \mathbf{I}_n) = \\
&= \sum_{i=1}^{n} \left(\frac{d}{d\lambda} \left(a_{i,i} - \lambda \right) \right) \cdot \mathrm{cof}_{i,i}(A - \lambda \cdot \mathbf{I}_n) = - \sum_{i=1}^{n} \mathrm{cof}_{i,i}(A - \lambda \cdot \mathbf{I}_n) = \\
&= -\mathrm{tr}(\mathrm{adj}(A - \lambda \cdot \mathbf{I}_n)).
\end{aligned}$$

Thus,

$$\mathrm{ch}'(\rho) = -\mathrm{tr}(\mathrm{adj}(A - \rho \cdot \mathbf{I}_n)).$$

Now, from the elementary properties of the adjugate matrix and from the fact that ρ is a root of $|A - \lambda \cdot \mathbf{I}_n|$, we obtain

$$(A - \rho \cdot \mathbf{I}_n) \cdot \mathrm{adj}(A - \rho \cdot \mathbf{I}_n) = |A - \rho \cdot \mathbf{I}_n| \cdot \mathbf{I}_n = \mathbf{0}. \tag{A.23}$$

By Lemma A.4.15, the Perron-Frobenius eigenvalue ρ has a geometric multiplicity equal to 1, i.e., the eigenspace corresponding to ρ has dimension 1. Thus, the rank of the matrix $A - \rho \cdot \mathbf{I}_n$ is $n - 1$. As a consequence, at least one cofactor of $A - \rho \cdot \mathbf{I}_n$ is nonzero, and thus

$$\mathrm{adj}(A - \rho \cdot \mathbf{I}_n) \neq \mathbf{0}.$$

The adjugate matrix $\mathrm{adj}(A - \rho \cdot \mathbf{I}_n)$ therefore has at least one nonzero column. Let i in $\{1, \ldots, n\}$ be a number of some nonzero column of $\mathrm{adj}(A - \rho \cdot \mathbf{I}_n)$, and let $\mathbf{c}_i$ be the corresponding column vector.

By (A.23), we get $(A - \rho \cdot \mathbf{I}_n) \cdot \mathbf{c}_i = \mathbf{0}$. Thus, $\mathbf{c}_i$ is an eigenvector corresponding to ρ. Since ρ is of geometric multiplicity 1 and by Theorem A.4.14, ρ has a positive eigenvector, $\mathbf{c}_i$ has to be a nonzero real multiple of a positive column vector, i.e., the entries of $\mathbf{c}_i$ are either all strictly positive, or all strictly negative.

Since $\mathbf{c}_i$ has been chosen as an arbitrary nonzero column of $\mathrm{adj}(A - \rho \cdot \mathbf{I}_n)$, it follows that each column of $\mathrm{adj}(A - \rho \cdot \mathbf{I}_n)$ is either strictly positive or strictly negative or zero. Moreover, there is at least one nonzero column of $\mathrm{adj}(A - \rho \cdot \mathbf{I}_n)$.

Further, $\mathrm{adj}(A - \rho \cdot \mathbf{I}_n)^T = \mathrm{adj}(A^T - \rho \cdot \mathbf{I}_n)$, where A^T is an irreducible nonnegative matrix with the Perron-Frobenius eigenvalue ρ. Thus, we may conclude the same for the columns of $\mathrm{adj}(A - \rho \cdot \mathbf{I}_n)^T$, i.e., each column of $\mathrm{adj}(A - \rho \cdot \mathbf{I}_n)^T$ is either strictly positive or strictly negative or zero, and there is at least one nonzero column of $\mathrm{adj}(A - \rho \cdot \mathbf{I}_n)^T$. In other words, each row of $\mathrm{adj}(A - \rho \cdot \mathbf{I}_n)$ is either strictly positive or strictly negative or zero, and there is at least one nonzero row of $\mathrm{adj}(A - \rho \cdot \mathbf{I}_n)$.

As we have already observed, $\mathrm{adj}(A - \rho \cdot \mathbf{I}_n)$ has at least one nonzero entry, say, (i, j). Then the whole i-th row must be strictly positive or strictly negative, which implies that either all columns of $\mathrm{adj}(A - \rho \cdot \mathbf{I}_n)$ are strictly positive or all columns are strictly negative. Thus, $\mathrm{adj}(A - \rho \cdot \mathbf{I}_n)$ is a positive matrix or a negative matrix. Therefore,

$$\mathrm{ch}'(\rho) = -\mathrm{tr}(\mathrm{adj}(A - \rho \cdot \mathbf{I}_n)) \neq 0,$$

i.e., the algebraic multiplicity of ρ is 1. □

We shall end up this subsection by a simple corollary of Lemma A.4.15, which assures that the Perron-Frobenius eigenvector of a given irreducible nonnegative matrix is unique.

Corollary A.4.17 Let A be an $n \times n$ irreducible nonnegative matrix, for some n in $\mathbb{N}$. Let ρ be the Perron-Frobenius eigenvalue of the matrix A. Then there is exactly one eigenvector $\mathbf{x}_\rho$ in E^n, corresponding to ρ. Moreover, the Collatz-Wielandt function $f_A(\mathbf{x})$ attains exactly one maximum in E^n.

Proof. Theorem A.4.14 assures the existence of at least one such eigenvector $\mathbf{x}'_\rho$. However, by Lemma A.4.15, the dimension of the eigenspace corresponding to ρ is 1, and thus each eigenvector corresponding to ρ has to be a scalar multiple of $\mathbf{x}'_\rho$. However, for $c \neq 1$, $c \cdot \mathbf{x}'_\rho$ cannot be in E^n. Thus, the eigenvector $\mathbf{x}_\rho = \mathbf{x}'_\rho$ is unique.

The existence of a maximum of $f_A(\mathbf{x})$ in E^n is assured by Theorem A.4.13. However, in Theorem A.4.14 we have proved that every such maximum is an eigenvector of A corresponding to ρ. Thus, this maximum is unique as well. □

A.4.4 Dominating Nonnegative Matrix

In this subsection, we shall present a definition of a *dominating nonnegative matrix* and prove an important theorem (Theorem A.4.19) due to Wielandt [40] pointing out an interesting property of these matrices. We shall use the presented theorem in the proof of the Perron-Frobenius theorem in Section A.4.5. As in the previous subsections, we shall follow the presentation of [31].

Definition A.4.18 Let $C = \left(c_{i,j}\right)_{n\times n}$ in $\mathbb{C}^{n\times n}$ be a complex matrix, and $A = \left(a_{i,j}\right)_{n\times n}$ in $\mathbb{P}^{n\times n}$ be a nonnegative real matrix, for some n in $\mathbb{N}$. The nonnegative matrix A is said to *dominate* the complex matrix C, if for all i, j in $\{1,\ldots,n\}$, the inequality $|c_{i,j}| \le a_{i,j}$ holds.

In other words, a nonnegative matrix A dominates the complex matrix C, if each entry of A is greater than or equal to the absolute value of the corresponding entry of C.

Theorem A.4.19 (Wielandt [40]) Let $C = \left(c_{i,j}\right)_{n\times n}$ in $\mathbb{C}^{n\times n}$ be a complex matrix, and $A = \left(a_{i,j}\right)_{n\times n}$ in $\mathbb{P}^{n\times n}$ be an irreducible nonnegative matrix dominating C. Let ρ be the Perron-Frobenius eigenvalue of the matrix A, and λ be an arbitrary eigenvalue of C. Then the inequality

$$|\lambda| \le \rho \tag{A.24}$$

holds. Moreover, the equality occurs in (A.24), if and only if a real number φ in $\mathbb{R}$ exists, such that $\lambda = \rho \cdot e^{i\varphi}$, and

$$C = e^{i\varphi} \cdot D \cdot A \cdot D^{-1}$$

for some complex diagonal matrix $D = (d_{i,j})_{n\times n}$ in $\mathbb{C}^{n\times n}$, such that $|d_{i,i}| = 1$ for $i = 1,\ldots,n$.

Proof. Let $\mathbf{y}$ be an eigenvector of the matrix C corresponding to the eigenvalue λ, i.e., a column vector $\mathbf{y} = (y_1,\ldots,y_n)^T \ne \mathbf{0}$, such that

$$C \cdot \mathbf{y} = \lambda \cdot \mathbf{y}. \tag{A.25}$$

By the triangle inequality, we may prove (in exactly the same way as in the proof of Theorem A.4.14) that

$$|C| \cdot |\mathbf{y}| \ge |\lambda| \cdot |\mathbf{y}|, \tag{A.26}$$

where $|C|$ is the matrix $(|c_{i,j}|)_{n\times n}$, and $|\mathbf{y}|$ is the column vector $(|y_1|,\ldots,|y_n|)^T$. Now, since the matrix A dominates C, and since all entries of $|\mathbf{y}|$ are nonnegative, (A.26) implies

$$A \cdot |\mathbf{y}| \ge |C| \cdot |\mathbf{y}| \ge |\lambda| \cdot |\mathbf{y}|, \tag{A.27}$$

and thus, consecutively by the claim 2 of Lemma A.4.11, by the claim 1 of Lemma A.4.11, and by Theorem A.4.14,

$$|\lambda| \le f_A(|\mathbf{y}|) = f_A\left(\frac{1}{\sum_{i=1}^n |y_i|} \cdot |\mathbf{y}|\right) \le \rho, \tag{A.28}$$

and the first part of the theorem is proved.

Now, let us prove the second part of the theorem. For the proof of the first implication, let us suppose that $C = e^{i\varphi} \cdot D \cdot A \cdot D^{-1}$ for some φ in $\mathbb{R}$, and for some complex diagonal matrix $D = (d_{i,j})_{n\times n}$ in $\mathbb{C}^{n\times n}$, such that $|d_{i,i}| = 1$ for $i = 1,\ldots,n$. The matrices C and $e^{i\varphi} \cdot A$ are then similar, i.e., $\lambda = \rho \cdot e^{i\varphi}$ is an eigenvalue of C, and clearly $|\lambda| = \rho$.

Let us now prove the converse implication. For this purpose, let us suppose that in (A.24), the equality holds, i.e., $|\lambda| = \rho$. Then, a real number φ exists, such that $\lambda = \rho \cdot e^{i\varphi}$. Then from (A.28), we get $f_A(|\mathbf{y}|) = \rho$. Thus, according to Theorem A.4.14,

$$|\mathbf{y}| = \left(\sum_{i=1}^n |y_i|\right) \cdot \left(\frac{1}{\sum_{i=1}^n |y_i|} \cdot |\mathbf{y}|\right) = \left(\sum_{i=1}^n |y_i|\right) \cdot \mathbf{x}_\rho$$

is a positive eigenvector corresponding to ρ (i.e., a nonzero scalar multiple of the Perron-Frobenius eigenvector). Therefore, (A.27) implies

$$A \cdot |\mathbf{y}| = |C| \cdot |\mathbf{y}| = \rho \cdot |\mathbf{y}|, \tag{A.29}$$

i.e.,

$$(A - |C|) \cdot |\mathbf{y}| = \mathbf{0}. \tag{A.30}$$

Since $|\mathbf{y}|$ is a nonnegative column vector and, at the same time, a nonzero scalar multiple of the Perron-Frobenius eigenvector, $|\mathbf{y}|$ must be positive (since, by Theorem A.4.14, the Perron-Frobenius eigenvector is positive). Moreover, since A dominates C, $A - |C|$ is a nonnegative matrix, and thus (A.30) implies

$$A = |C|. \tag{A.31}$$

Now, let us define the diagonal matrix D by

$$D = \begin{pmatrix} \frac{y_1}{|y_1|} & & & \\ & \frac{y_2}{|y_2|} & & \\ & & \ddots & \\ & & & \frac{y_n}{|y_n|} \end{pmatrix}.$$

Moreover, let us define

$$G = (g_{i,j})_{n\times n} = e^{-i\varphi} \cdot D^{-1} \cdot C \cdot D.$$

Since clearly $\mathbf{y} = D \cdot |\mathbf{y}|$, from (A.25) we obtain

$$C \cdot D \cdot |\mathbf{y}| = \lambda \cdot D \cdot |\mathbf{y}| = \rho \cdot e^{i\varphi} \cdot D \cdot |\mathbf{y}|.$$

Thus,

$$G \cdot |\mathbf{y}| = \rho \cdot |\mathbf{y}|,$$

and by (A.29),

$$G \cdot |\mathbf{y}| = A \cdot |\mathbf{y}|. \tag{A.32}$$

Now, G was defined so that $|G| = |C|$ holds (where, as in the previous cases, $|G|$ is defined to be a matrix $(|g_{i,j}|)_{n\times n}$), and therefore, by (A.31),

$$|G| = A.$$

Thus, (A.32) implies

$$|G| \cdot |\mathbf{y}| = G \cdot |\mathbf{y}|.$$

That is,

$$(|G| - G) \cdot |\mathbf{y}| = \mathbf{0}$$

i.e.,

$$\sum_{j=1}^{n} (|g_{i,j}| - g_{i,j}) \cdot |y_j| = 0, \qquad i = 1, \dots, n.$$

As we have already noted, the vector $|\mathbf{y}|$ is positive, and therefore, $|y_j| > 0$ for $j = 1, \dots, n$. Thus, since the real part of $|g_{i,j}| - g_{i,j}$ is clearly nonnegative for $i, j = 1, \dots, n$, $|g_{i,j}| - g_{i,j} = 0$ for $i, j = 1, \dots, n$. Thus, we have proved that

$$G = |G| = A.$$

Thus, from the definition of G, we have

$$C = e^{i\varphi} \cdot D \cdot G \cdot D^{-1} = e^{i\varphi} \cdot D \cdot A \cdot D^{-1}$$

and the theorem is proved. □

A.4.5 The Perron-Frobenius Theorem

Now we are prepared to state and prove the main result of this section, i.e., the *Perron-Frobenius theorem*. The core of the proof is by [31].

Theorem A.4.20 (Perron, Frobenius [33] [13]) Let $A = \left(a_{i,j}\right)_{n\times n}$ be an $n \times n$ irreducible nonnegative matrix, for some n in $\mathbb{N}$. Let $\lambda := \rho(A)$ be the spectral radius of the matrix A. Then a positive integer p in $\mathbb{N}^+$ exists, such that the following properties hold:

1. The matrix A has exactly p complex eigenvalues of absolute value λ, which are
$$\lambda, \lambda \cdot e^{2\pi i/p}, \ldots, \lambda \cdot e^{2\pi i(p-1)/p}.$$
All of these eigenvalues have their algebraic multiplicity equal to 1.

2. The (right) eigenvector $\mathbf{x}$ of the matrix A corresponding to the eigenvalue λ is positive. Moreover, all nonnegative eigenvectors of the matrix A are scalar multiples of $\mathbf{x}$.

3. The matrix A is similar to $e^{2\pi i/p} \cdot A$, i.e., the spectrum of the matrix A is invariant under the multiplication by $e^{2\pi i/p}$.

Proof. In Theorem A.4.14, we have proved that λ is an eigenvalue of the matrix A, and in Theorem A.4.16, we have proved that the algebraic multiplicity of λ is 1. Now, let us suppose that the matrix A has p eigenvalues of absolute value λ,
$$\lambda_1, \lambda_2, \ldots, \lambda_p.$$
Without loss of generality, let us suppose that $\lambda_1 = \lambda$, and $\lambda_2, \ldots, \lambda_p \neq \lambda$. Then, each of the eigenvalues λ_j, $j = 2, \ldots, p$, can be written as
$$\lambda_j = \lambda \cdot e^{i\varphi_j},$$
for some nonzero φ_j in $\mathbb{R}$. Moreover, the matrix A clearly dominates itself. Thus, by Theorem A.4.19, a complex diagonal matrix D_j exists for $j = 2, \ldots, p$, such that
$$A = e^{i\varphi_j} \cdot D_j \cdot A \cdot D_j^{-1}. \tag{A.33}$$
In other words, the matrices A and $e^{i\varphi_j} \cdot A$ are similar, and therefore have the same spectrum. Moreover, since λ is a simple eigenvalue of A, $\lambda \cdot e^{i\varphi_j}$ is a simple eigenvalue of $e^{i\varphi_j} \cdot A$. That is, for $j = 2, \ldots, p$, $\lambda \cdot e^{i\varphi_j}$ is a simple eigenvalue of A. Since λ is real, let $\varphi_1 = 0$.

Without loss of generality, let us suppose $0 = \varphi_1 < \varphi_2 < \ldots < \varphi_p < 2\pi$. We shall show that
$$\varphi_j = \frac{2\pi(j-1)}{p}, \tag{A.34}$$
for $j = 1, \ldots, p$. To achieve this, let r, s be two (not necessarily distinct) numbers in $\{2, \ldots, n\}$. Then, by (A.33),
$$\begin{aligned} A = e^{i\varphi_r} \cdot D_r \cdot A \cdot D_r^{-1} &= e^{i\varphi_r} \cdot D_r \cdot \left(e^{i\varphi_s} \cdot D_s \cdot A \cdot D_s^{-1}\right) \cdot D_r^{-1} = \\ &= e^{i(\varphi_r+\varphi_s)} \cdot (D_r \cdot D_s) \cdot A \cdot (D_r \cdot D_s)^{-1}, \end{aligned}$$
i.e., the matrices A and $e^{i(\varphi_r+\varphi_s)} \cdot A$ are similar for all r, s in $\{2, \ldots, p\}$ and thus for all r, s in $\{1, \ldots, p\}$. Thus, for all r, s in $\{1, \ldots, p\}$, $\lambda \cdot e^{i(\varphi_r+\varphi_s)}$ is an eigenvalue of A, and since its absolute value is λ, the equality
$$\lambda \cdot e^{i(\varphi_r+\varphi_s)} = \lambda \cdot e^{i\varphi_t} = \lambda_t$$

has to hold for some t in $\{1, \ldots, p\}$. It can be easily seen that this implies that $e^{i\varphi_1}, \ldots, e^{i\varphi_p}$ are distinct p-th roots of unity, and from our assumptions about the order of $\varphi_1, \ldots, \varphi_p$, (A.34) follows.

Now, let us prove the claim 2. The existence of the positive eigenvector $\mathbf{x} = (x_1, \ldots, x_n)^T$ corresponding to λ follows directly from Theorem A.4.14. Thus, it remains to show the second part of the claim, i.e., that every nonnegative eigenvector of A is a scalar multiple of $\mathbf{x}$.

To achieve this, it is clearly sufficient to show that the matrix A has exactly one eigenvector in E^n (see Denotation A.4.12). In fact, let $\mathbf{y}$ be a positive column vector in E^n defined to be the Perron-Frobenius eigenvector of A^T. Since the spectrum of A^T is the same as the spectrum of A, $\mathbf{y}$ corresponds to the eigenvalue λ.

Now, let $\mathbf{z}$ in E^n be an eigenvector of the matrix A corresponding to an eigenvalue ζ, i.e., $A \cdot \mathbf{z} = \zeta \cdot \mathbf{z}$. If we denote the standard inner product on $\mathbb{R}^{n \times 1}$ by $\langle \cdot, \cdot \rangle$, we obtain

$$\zeta \cdot \langle \mathbf{z}, \mathbf{y} \rangle = \langle A \cdot \mathbf{z}, \mathbf{y} \rangle = \langle \mathbf{z}, A^T \cdot \mathbf{y} \rangle = \lambda \cdot \langle \mathbf{z}, \mathbf{y} \rangle. \tag{A.35}$$

However, $\mathbf{y}$ is positive and $\mathbf{z}$ is in E^n, and therefore $\langle \mathbf{z}, \mathbf{y} \rangle > 0$. Thus, (A.35) implies $\zeta = \lambda$, i.e., $\mathbf{z}$ is an eigenvector of A corresponding to λ. The uniqueness of $\mathbf{z}$ then follows from Lemma A.4.15.

The claim 3 follows from (A.33) for $j = 2$. The theorem is therefore proved. □

Remark A.4.21 The Perron-Frobenius theorem obviously holds also for the 1×1 zero matrix, which is not considered to be irreducible in this report (although is considered to be irreducible, e.g., in [31]). Thus, we shall extend the notion of the Perron-Frobenius eigenvalue also to this matrix (clearly, this eigenvalue is zero).

A.4.6 The Normal Form of a Reducible Matrix

The Perron-Frobenius theorem, presented in Subsection A.4.5, provides us with a useful partial characterization of eigenvalues of irreducible matrices. However, as we have already mentioned, in this report we are interested in the theory of nonnegative matrices mainly for the reason that transition matrices of deterministic finite automata are nonnegative. But a transition matrix of a deterministic finite automaton need not to be irreducible. Thus, the need arises to study the eigenvalues of nonnegative matrices in general, without the assumption of irreducibility.

As we shall show in this subsection, a useful partial characterization of eigenvalues of nonnegative matrices in general can be obtained by using the concept of the *normal form of a reducible matrix*. We shall show that a spectrum of a general nonnegative matrix is in fact a union of spectra of irreducible matrices that can be found easily for a given nonnegative matrix.

In this report, we use the definition of [39], i.e., the normal form of a reducible matrix is an upper-triangular block matrix. However, in [31], the normal form of a reducible matrix is defined to be a lower-triangular block matrix.

Definition A.4.22 Let A be an $n \times n$ nonnegative matrix, for some n in $\mathbb{N}$. The matrix A is said to be in the *normal form of a reducible matrix*, if it has a block form

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & \ldots & A_{1,m} \\ \mathbf{0} & A_{2,2} & \ldots & A_{2,m} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & A_{m,m} \end{pmatrix},$$

where m is in $\mathbb{N}$, and $A_{i,i}$ are square submatrices for $i = 1, \ldots, m$ that are either irreducible or 1×1 null matrices.

Let us note that talking about the normal form of a *reducible* matrix may be quite misleading, since in fact every irreducible matrix satisfies the normal form. That is, talking about the normal form of a *nonnegative* matrix would probably be more accurate.

The concept of the normal form of a reducible matrix has a connection to graph theory. If a digraph G has an adjacency matrix A, it is clear that a permutation matrix P exists, such that the matrix $P \cdot A \cdot P^{-1}$ is in the normal form: it suffices to topologically sort the strongly connected components of the digraph G and choose a permutation matrix P such that rows and columns of $P \cdot A \cdot P^{-1}$ corresponding to vertices of a given strongly connected component C have lower indices than the rows and columns corresponding to vertices of each strongly connected component that is after C in the topological order. Each diagonal block of a matrix in the normal form of a reducible matrix therefore corresponds to one strongly connected component of the corresponding digraph. Thus, it is intuitively clear that the following theorem holds.

Theorem A.4.23 Let A be an $n \times n$ nonnegative matrix, for some n in $\mathbb{N}$. Then an $n \times n$ permutation matrix P exists, such that the matrix

$$P \cdot A \cdot P^{-1}$$

is in the normal form of a reducible matrix.

Proof. By induction on n.

1. Let $n = 1$. Then the matrix A is either irreducible or 1×1 null, and thus A is in the normal form of a reducible matrix.

2. Let us suppose that the theorem holds for all $n \leq k - 1$ for some $k \geq 2$. We shall show that it holds also for $n = k$. Let A be a $k \times k$ nonnegative matrix. Then it is either irreducible or reducible. If it is irreducible, then it is in the normal form of a reducible matrix and a theorem holds.

   Let us therefore suppose that A is reducible. Then, by the definition of reducible matrices, square matrices B and D exist, such that

   $$P \cdot A \cdot P^{-1} = \begin{pmatrix} B & C \\ \mathbf{0} & D \end{pmatrix},$$

   for some permutation matrix P and submatrix C. But B and D are nonnegative matrices of size smaller than is the size of A and therefore, by the induction hypothesis, permutation matrices P_1, P_2 exist, such that

   $$P_1 \cdot B \cdot P_1^{-1}$$

   and

   $$P_2 \cdot D \cdot P_2^{-1}$$

   are in the normal form of a reducible matrix. Thus, clearly,

   $$\begin{pmatrix} P_1 & \mathbf{0} \\ \mathbf{0} & P_2 \end{pmatrix} \cdot P \cdot A \cdot P^{-1} \cdot \begin{pmatrix} P_1 & \mathbf{0} \\ \mathbf{0} & P_2 \end{pmatrix}^{-1} = \left(\begin{pmatrix} P_1 & \mathbf{0} \\ \mathbf{0} & P_2 \end{pmatrix} \cdot P \right) \cdot A \cdot \left(\begin{pmatrix} P_1 & \mathbf{0} \\ \mathbf{0} & P_2 \end{pmatrix} \cdot P \right)^{-1}$$

   is in the normal form of a reducible matrix.

The theorem is proved. □

In other words, we have proved that each matrix is similar to a matrix in the normal form of a reducible matrix. Moreover, the proof of Theorem A.4.23 provides us with a method of finding this matrix. Therefore, for a given nonnegative matrix, to find eigenvalues with the maximal absolute value it is sufficient to find this normal form and the Perron-Frobenius eigenvalues of its diagonal blocks.

A.5 Three Number-Theoretic Lemmas

In this section, we shall present three number-theoretic lemmas, third of which we use in the study of $\mathcal{S}$-equiloaded deterministic finite automata, in Section 2.3 of Chapter 2 (the former two are necessary for the proof of the third lemma). The presented lemmas are also of great use in the theory of nonnegative matrices. For instance, they can be used to prove an extremely useful characterization of primitive matrices (however, we are not concerned with this kind of nonnegative matrices in Section A.4). More generally, we can say that the presented lemmas are useful in situations, when there is a need to analyze the behaviour of cycles in directed graphs (that are closely related to nonnegative matrices as well as deterministic finite automata). We shall follow the presentation of [3].

Lemma A.5.1 Let $S \subseteq \mathbb{Z}$ be a set of integers containing at least one nonzero element $c \neq 0$. Moreover, let S be closed under addition and under subtraction. Then a positive integer a in $\mathbb{N}^+$ exists, such that $S = \{ka \mid k \in \mathbb{Z}\}$.

Proof. Since S is closed under subtraction and $c \neq 0$ is in S, also $0 = c - c$ and $-c = 0 - c$ are in S. Thus, S contains both c and $-c$, which implies that S contains at least one positive element. Let a be the smallest positive element in S. Since S is closed under addition, the elements $a, 2a = a + a, 3a = 2a + a, \ldots$ are all in S. Moreover, since S is closed under subtraction, the elements $-a = 0 - a, -2a = -a - a, -3a = -2a - a, \ldots$ are all in S. Thus $S \supseteq \{ka \mid k \in \mathbb{Z}\}$.

Now we shall prove $S \subseteq \{ka \mid k \in \mathbb{Z}\}$. Let b be in A. Then $b = ka + r$, where k is in $\mathbb{Z}$, and r is in $\{0, 1, \ldots, a-1\}$. But since S is closed under subtraction, $b - ka = r$ is also in S. If r was in $\{1, \ldots, a-1\}$, then a would not be the smallest positive element of S, i.e., we would obtain a contradiction. Thus, $r = 0$ and $b = ka$ for some k in $\mathbb{Z}$. Since this holds for all b in A, the inclusion $S \subseteq \{ka \mid k \in \mathbb{Z}\}$ has to hold. □

Lemma A.5.2 (Bézout's lemma) Let $a_1, \ldots, a_k$ in $\mathbb{N}^+$ be positive integers with greatest common divisor d. Then integers $n_1, \ldots, n_k$ in $\mathbb{Z}$ exist,[19] such that

$$d = \sum_{i=1}^{k} n_i a_i.$$

Proof. Let us define the set S by

$$S = \left\{ \sum_{i=1}^{k} m_i a_i \;\middle|\; m_1, \ldots, m_k \in \mathbb{Z} \right\}.$$

Let b and c be in S. Then

$$b = \sum_{i=1}^{k} m_i^{(b)} a_i$$

and

$$c = \sum_{i=1}^{k} m_i^{(c)} a_i$$

for some $m_1^{(b)}, \ldots, m_k^{(b)}, m_1^{(c)}, \ldots, m_k^{(c)}$ in $\mathbb{Z}$. Thus,

$$b + c = \sum_{i=1}^{k} \left(m_i^{(b)} + m_i^{(c)} \right) a_i$$

and

$$b - c = \sum_{i=1}^{k} \left(m_i^{(b)} - m_i^{(c)} \right) a_i,$$

[19] The values of $n_1, \ldots, n_k$ can be found by the *Extended Euclidean algorithm* (see, e.g., [8]).

i.e., $b+c$ and $b-c$ are also in S. That is, the set S is closed under addition and under subtraction. Therefore, according to Lemma A.5.1, a positive integer a in $\mathbb{N}^+$ exists, such that S can be written as $S = \{ka \mid k \in \mathbb{Z}\}$. Let

$$a = \sum_{i=1}^{k} m_i^{(a)} a_i$$

for some $m_1^{(a)}, \ldots, m_k^{(a)}$ in $\mathbb{Z}$.

Now, d is the greatest common divisor of $a_1, \ldots, a_k$, and therefore d divides each a_i, $i = 1, \ldots, k$. Thus d divides a, which implies $0 < d \leq a$. Moreover, $a_1, \ldots, a_k$ are clearly all in S, and since $S = \{ka \mid k \in \mathbb{Z}\}$, a divides each a_i, $i = 1, \ldots, k$. Therefore a is a common divisor of $a_1, \ldots, a_k$, and thus $a \leq d$, since d is their *greatest* common divisor. Thus, we have $d \leq a$ and $a \leq d$, i.e., $a = d$. Thus, d is in S and integers $n_1, \ldots, n_k$ in $\mathbb{Z}$ exist such that

$$d = \sum_{i=1}^{k} n_i a_i.$$

The lemma is therefore proved. □

Lemma A.5.3 Let S be a set of positive integers, satisfying the following two conditions:

(i) The greatest common divisor of the elements of S is d.

(ii) The set S is closed under addition.

Then a positive integer n_0 exists, such that for all positive integers $n \geq n_0$, such that d divides n, n is in S.

Proof. Let us first assume that $d = 1$. Every set of positive integers is countable, and therefore the elements of S can be ordered into a sequence $\{a_n\}_{n=1}^{\infty}$. Let us define the function $f(n)$ to be the greatest common divisor of $a_1, \ldots, a_n$ for all $\mathbb{N}^+$. The function $f(n)$ is clearly non-increasing and bounded below by 1. Therefore, $f(n)$ has a limit (the greatest common divisor[20] of elements of S). Since the values of $f(n)$ are integers, the limit has to be attained after a finite number of steps. Therefore, a positive integer k in $\mathbb{N}^+$ exists, such that $f(n)$ is the greatest common divisor of elements of S, for all $n \geq k$.

Since the greatest common divisor of elements of S is 1, there is a positive integer k in $\mathbb{N}^+$, such that the greatest common divisor of $a_1, \ldots, a_k$ is 1. Therefore, by Lemma A.5.2, integers $n_1, \ldots, n_k$ in $\mathbb{Z}$ exist, such that

$$1 = \sum_{i=1}^{k} n_i a_i. \tag{A.36}$$

Let us define

$$P_+ = \sum_{\substack{1 \leq i \leq k \\ n_i > 0}} n_i a_i$$

to be the sum of the positive elements of the sum in (A.36), and

$$P_- = -\sum_{\substack{1 \leq i \leq k \\ n_i < 0}} n_i a_i$$

[20] It is clear that if the limit exists, it has to be equal to the value of the greatest common divisor. Formally, it can be proved as follows: since all function values $f(n)$ are positive integers, the limit has to be a positive integer as well. Now, if the limit L is smaller than the greatest common divisor, there has to be n such that $f(n) = L$, and that is the contradiction. On the other hand, if the limit L is greater than the greatest common divisor, there must be a positive integer m, such that a_m is not divisible by L (if not, L would be a common divisor greater than the greatest common divisor, i.e., a contradiction). But this implies $f(m) < L$, and since $f(n)$ is non-increasing, $f(n) < L$ for all $n \geq m$. But then the function $f(n)$ cannot have L as the limit.

to be the sum of the negative elements (made positive). From (A.36), we clearly have $1 = P_+ - P_-$. Clearly, both P_+ and P_- are in S.

Now, let n in $\mathbb{N}^+$ be an arbitrary positive integer, such that $n \geq P_-(P_- - 1)$. As well as any other positive integer, n can be written as

$$n = aP_- + r,$$

where a is in $\mathbb{N}$ and r is in $\{0, 1, \ldots, P_- - 1\}$. Clearly, $a \geq P_- - 1$ (otherwise, we would have $a \leq P_- - 2$, and therefore $n = aP_- + r < P_-(P_- - 1)$). Now, since $1 = P_+ - P_-$, we have

$$n = aP_- + r = aP_- + r(P_+ - P_-) = (a - r)P_- + rP_+.$$

But since $a \geq P_- - 1$, we also have the inequality $a \geq r$, i.e., $a - r \geq 0$. Since $n \geq P_-(P_- - 1) > 0$, at least one of the numbers $a - r$ and r is positive. Thus, since P_- and P_+ are both in S, both $a - r$ and r are nonnegative and at least one of them is positive, from the fact that S is closed under addition, we may conclude that n is in S. Thus, we have proved the statement of the lemma for $n_0 = P_-(P_- - 1)$.

Now, let $d > 1$. Then, if we divide all elements of S by d, we obtain the set S' that is closed under addition and the greatest common divisor of its elements is 1. Thus, by what we have proved above, the positive integer n'_0 in $\mathbb{N}^+$ exists, such that for all $n \geq n'_0$, n is in S'. Thus, if we multiply all elements of S' by d, the resulting set will be clearly equal to S and the property from the statement of the lemma will clearly hold. Thus, the lemma is proved. $\square$