The Formal Language Theory Column

by

Giovanni Pighizzini

Dipartimento di Informatica
Università degli Studi di Milano
20133 Milano, Italy
pighizzini@di.unimi.it

25 EDITIONS OF DCFS: ORIGINS AND DIRECTIONS

Jürgen Dassow
Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
PSF 4120, 39016 Magdeburg, Germany
dassow@iws.cs.uni-magdeburg.de

Martin Kutrib
Institut für Informatik, Universität Giessen
Arndtstr. 2, 35392 Giessen, Germany
kutrib@informatik.uni-giessen.de

Giovanni Pighizzini
Dipartimento di Informatica
Università degli Studi di Milano
Via Celoria, 18, 20133 Milano, Italy
pighizzini@di.unimi.it

Abstract

Since the late nineties the scope of the *International Conference of Descriptional Complexity of Formal Systems* (DCFS) encompasses all aspects of descriptional complexity, both in theory and application. We first consider the historical development of the conference. Then we turn to some impressions from the 25 editions of the conference, which we particularly remember. In order to give a deeper inside in the field of descriptional complexity, we present some of its very basics from a general abstract perspective. Then we turn to some of the outstanding and dominating directions in the course of time. The results presented are not proved but we merely draw attention to the overall picture and some of the main ideas involved.

1 Introduction

Since the dawn of theoretical computer science the relative succinctness of different representations of (sets of) objects by formal systems have been a subject of intensive research. An obvious choice to encode the objects is by strings over a finite alphabet. Then a set of objects is a set of strings, that is, a formal language. Formal languages can be described by several means, for example, by automata, grammars, rewriting systems, equation systems, etc. In general, such a descriptional system is a set of finite descriptors for languages. Core questions of descriptional complexity are "How succinctly (related to a size complexity measure) can a system represent a formal language in comparison with other systems?" and "What is the maximum trade-off when the representation is changed from one descriptional system to another, and can this maximum be achieved?" In the classification of automata, grammars, and related (formal) systems it turned out that the gain in economy of description heavily depends on the considered systems.

The approach to analyze the size of systems as opposed to the computational power seems to originate from Stearns [115] who studied the relative succinctness of regular languages represented by deterministic finite automata (DFAs) and deterministic pushdown automata. He showed the decidability of regularity for deterministic pushdown automata in a deep proof. The effective procedure revealed the following upper bound for the simulation. Given a deterministic pushdown automaton with $n > 1$ states and $t > 1$ stack symbols that accepts a regular language, then the number of states which is sufficient for an equivalent DFA is bounded by an expression of the order $t^{n^{n^n}}$. Later this triple exponential upper bound has been improved by one level of exponentiation in [116]. In the levels of exponentiation it is tight, as proved in [95] by obtaining a double exponential lower bound. The precise bound is still an open problem. Probably the best-known result on descriptional complexity is the construction of a DFA that simulates a given nondeterministic finite automaton (NFA) [113]. By this so-called *powerset construction*, each state of the DFA is associated with a subset of NFA states. Moreover, the construction turned out to be optimal, in general. That is, the bound on the number of states necessary for the construction is tight in the sense that for an arbitrary n there is always some n-state NFA which cannot be simulated by any DFA with strictly less than 2^n states [79, 95, 97].

Let us turn to another cornerstone of descriptional complexity theory in the seminal paper by Meyer and Fischer [95]. In general, a known upper bound for the trade-off answers the question, how succinctly can a language be represented by a descriptor of one descriptional system compared with the representation by an equivalent descriptor of the other descriptional system? In [95] the sizes of finite automata and general context-free grammars for regular languages are compared. The comparison revealed a qualitatively new phenomenon. The gain in economy of description can be arbitrary, that is, there are no recursive functions serving as upper bounds for the trade-off, which is said to be *non-recursive*. Non-recursive trade-offs usually sprout at the wayside of the crossroads of (un)decidability, and in many cases proving such trade-offs apparently requires ingenuity and careful constructions.

Nowadays, descriptional complexity has become a large and widespread area. On our tour on the field we first consider the historical development of the conference *Descriptional Complexity of Formal Systems* (DCFS). Then we turn to some impressions from the 25 editions of the conference, which we particularly remember. In order to give a deeper inside in the field of descriptional complexity, we present some of its very basics from a general abstract perspective. Our tour on the subjects covers some outstanding and dominating topics. It obviously lacks completeness and it reflects our personal view of what constitute some of the most interesting links to descriptional complexity theory. In truth there is much more to the field than can be summarized here and in the related papers [31, 39, 69, 70]. The results presented are not proved but we merely draw attention to the overall picture and some of the main ideas involved.

2 History of DCFS [1]

In 1998, the history of DCFS started at the conference *Mathematical Foundations of Computer Science* (MFCS) in Brno. During a lunch break, Detlef Wotschke (1944–2019) suggested the organization of a workshop on descriptional complexity and related topics. It seems that there were two reasons for such a proposal.

Firstly, in 1997, within the organization *International Federation of Information Processing* (IFIP), a reestablishment of the Technical Committee TC1 *Foundations of Computer Science* took place, and within TC1 a Working Group WG 1.02 *Descriptional Complexity* was created. The chairman of this WG was Detlef Wotschke. It was natural to found a special workshop of the working group.

Secondly, the MFCS conference in Brno was accompanied by more than 10 workshops, some of them were organized as single events and some of them took place as a part of certain workshop series. Descriptional complexity was present in some of these workshops, but a special workshop on this topic was missing.

Detlef did not only come with the proposal of a workshop, he also had an idea for the place – Magdeburg (where DLT took place in 1995 and where Jürgen Dassow, the head of the group working in formal languages in Magdeburg, had a good position in the university). After some discussions, Jürgen accepted that his group will organize a workshop in Magdeburg in 1999.

In July 20–23, 1999, the workshop *Descriptional Complexity of Automata, Grammars and Related Systems* (DCAGRS) took place in Magdeburg. It was a

[1] The conference series *Descriptional Complexity of Formal Systems* has two roots, the workshops *Formal Descriptions and Software Reliability* and *Descriptional Complexity of Automata, Grammars and Related Systems*. Here we reflect only the latter one.

terrible title, but the organizers wanted a title which describes very well the topic of the workshop. The event was successful with respect to the invited lectures (e. g. J. Gruska, Sh. Yu (1950–2012), J. Shallit) as well as to the number and quality of submissions as well as to the large number of participants.

Detlef Wotschke
(1944–2019)

Helmut Jürgensen
(1942–2019)

We mention two facts where the first DCAGRS essentially differs from the later DCFS conferences. Firstly, the conference fee was only 70 euros, the average registration fee of some of the last normal DCFS conferences was 280 euros. Secondly, the program committee consisted of six person, the average of the last conferences was 24.

By the success of the first edition, it was necessary to look for a continuation. One weak before the workshop in Magdeburg, there was the *Workshop on Implementation of Automata* in Potsdam, organized by Helmut Jürgensen (1942–2019). During the excursion of this event (a boat tour through the lakes around Potsdam) Detlef and Jürgen talked to Helmut concerning the next DCAGRS. Finally, Helmut accepted to organize it in London (Ontario) (not knowing that it will be organized in London four times almost like a biannual conference, and that he will organize it three times). The site London was chosen, because, from the beginning, there was the idea to organize the workshop alternately in Europe and North America. This idea has been followed in the sequel with only three exceptions as one can see from Figure 1.

In the following years, there was a steering committee which was looking for the persons and places of the next (two) DCFS editions. This task was not easy in some cases, but finally the committee was successful in all the years. In 2015 the international workshop DCFS became an international conference to underline the grown importance and the history of the event. A list of all editions of DCFS is given in Figure 1.

	time	place	organizing institution / chairman
DCAGRS 1999	July 20–23	Magdeburg, Germany	O.-v.-Guericke-Universität Magdeburg J. Dassow
DCAGRS 2000	July 27–29	London, Canada	The University of Western Ontario H. Jürgensen
DCAGRS 2001	July 20–22	Vienna, Austria	Technical University of Vienna R. Freund
DCFS 2002	August 21–25	London, Canada	The University of Western Ontario H. Jürgensen
DCFS 2003	July 12–14	Budapest, Hungary	Hungarian Academy of Sciences E. Csuhaj-Varjú
DCFS 2004	July 26–28	London, Canada	The University of Western Ontario L. Ilie
DCFS 2005	June 30 - July 2	Como, Italy	University of Milan G. Pighizzini
DCFS 2006	June 21–23	Las Cruces, USA	New Mexico State University H. Leung
DCFS 2007	July 20–22	Nový Smokovec, Slovakia	R.J.Šafarik University Košice V. Geffert
DCFS 2008	July 16–18	Charlottetown, Canada	University of Prince Edward Island C. Câmpeanu
DCFS 2009	July 6–9	Magdeburg, Germany	O-v-Guericke-Universität Magdeburg J. Dassow, B. Truthe
DCFS 2010	August 8-10	Saskatoon, Canada	University of Saskatchewan I. McQuillan
DCFS 2011	July 25–27	Limburg, Germany	Justus-Liebig-Universität Gießen M. Holzer, M. Kutrib
DCFS 2012	July 23–25	Braga, Portugal	Univ. of Porto and Univ. of Minho N. Moreira, R. Reis
DCFS 2013	July 22-25	London, Canada	The University of Western Ontario H. Jürgensen
DCFS 2014	August 5–8	Turku, Finland	University of Turku J. Karhumäki, A. Okhotin
DCFS 2015	July 25–27	Waterloo, Canada	University of Waterloo J. Shallit
DCFS 2016	July 5–8	Bucharest, Romania	University of Bucharest C. Câmpeanu
DCFS 2017	July 3–5	Milan, Italy	University of Milan G. Pighizzini
DCFS 2018	July 25–27	Halifax, Canada	Saint Mary's University St. Konstantinides
DCFS 2019	July 17–19	Košice, Slovakia	Slovak Academy of Sciences G. Jirásková
DCFS 2020		Collected Papers	 G. Jirásková, G. Pighizzini
DCFS 2021		Collected Papers	 Y.-S.Han, S.-K. Ko
DCFS 2022	August, 29–31	Debrecen, Hungary	University of Debrecen Gy. Vaszil
DCFS 2023	July 4–6	Potsdam, Germany	University of Potsdam H. Bordihn

Figure 1: List of conferences.

We mention some important facts.

The initiative for DCFS came from the chairman of WG 1.02 of TC1 of IFIP, and all editions were organized by some institution and this IFIP working group together. Thus DCFS can be considered as the conference of WG 1.02. Since some years, the meetings of the Working Group take place as an evening session of DCFS.

If we compare the list of topics from 2007 (the oldest which can be found in the Web) and 2023 (the last conference) and those in between, then one notice that they are identical in appr. 75% of the items. This proves that there is a strong continuity and no following of short-lived *modern* directions.

In the years 2020 and 2021, DCFS conferences were planned in Vienna organized by R. Freund and in Seoul organized by Y.-S. Han and S.-K. Ko, respectively. Due to the crisis caused by the Corona virus, both conferences had to be canceled. However, there were invitations to the world for submitting papers such that proceedings could also be published in these years. Thanks go to the editors G. Jirásková/G. Pighizzini and Y.-S. Han/S.-K. Ko for their contribution to the survival of DCFS during the Corona time.

From the very beginning there was the idea that DCAGRS/DCFS should be organized with respect time and place in connection with some other conference such that e. g., only one crossing of the Atlantic Ocean is necessary to visit at least two conferences. As favorite accompanying conferences were considered *Developments in Language Theory* (DLT) and the *International Conference Implementation and Application of Automata* (CIAA, formerly *Workshop on Implementation of Automata*, WIA). Also this idea was realized for almost all editions (see Figure 2). Sometimes, the events were very near; for instance, in 2001, there was one day which was part of the DLT as well as of the DCFS program. Sometimes, the distance was large (in 2006, the distance between Las Cruces and Santa Barbara was 1600 km, but the Europeans had to cross the ocean only once; the four days between the two conferences could be used e. g., for a visit of the Grand Canyon almost in the middle between the towns).

The special event *50 Years of Automata Theory*, that took place in 2000, was particularly remarkable. The list of speakers was very impressive. One could hear, meet and talk to all those persons which contributed by famous basic theorems as M. Rabin, D. Scott, and Sh. Greibach, introduced essential concepts as R. McNaughton or wrote famous textbooks as J. Hopcroft and A. Salomaa (note that the mentioned names represent less than half of speakers). However, we do not know why automata theory became 50 years in 2000.

In 2015, one day before DCFS, the birthday of Janusz (John) Brzozowski (1935–2019) was celebrated in a one-day-conference.

Some remarks concerning proceedings. In the years 1999–2008, proceedings were published by the organizing institution. In the following two years, the pro-

year/place	accomp. event	year/place	accomp. event
1999 Magdeburg	WIA Potsdam	2010 Saskatoon	DLT London and
2000 London	CIAA London and		CIAA Winnipeg
	50 Years Automata Th.	2012 Braga	CIAA Porto
2001 Vienna	DLT Vienna	2013 London	CIAA Halifax
2003 Budapest	DLT Szeged	2014 Turku	CIAA Gießen
2004 London	CIAA Kingston	2015 Waterloo	Birthday Brzozowski
2005 Como	CIAA Sophia Antipolis	2017 Milan	DLT Liege and
2006 Las Cruces	DLT Santa Barbara		CIAA Marne-la-Vallé
2007 Nový Smokovec	CIAA Prague	2018 Halifax	CIAA Charlottetown
2008 Charlottetown	CIAA San Francisco	2019 Košice	CIAA Košice
2009 Magdeburg	DLT Stuttgart	2022 Debrecen	NCMA Debrecen

Figure 2: List of events accompanying DCAGRS/DCFS. (NCMA is an international workshop on *Non-Classical Models of Automata and Applications*.)

ceedings appeared in the series *Electronic Proceedings in Theoretical Computer Science* as numbers 3 and 31, respectively. There were some attempts to publish in the LNCS series of Springer-Verlag, but only in 2011 we were successful. Starting with the thirteenth edition of DCFS, the Proceedings appeared as *Lecture Notes in Computer Science*.

Proceedings have mostly a page limit for the contributions, i. e., they do not contain often full versions. Therefore, from the very beginning, full versions of selected papers were published as special issues of some scientific journals. Thus,

year DCFS	journal	volume (issue)	year journal	year DCFS	journal	volume (issue)	year journal
1999	JALC	5 (3)	2000	2012	JALC	17 (2–4)	2012
2000	JALC	6 (4)	2001	2013	IJFCS	25 (7)	2014
2001	JALC	7 (4)	2002	2014	TCS	610(A)	2016
2002	JALC	9 (2–3)	2004	2015	IC	259 (2)	2018
2003	TCS	330 (2)	2005	2016	JALC	22 (1–3)	2017
2004	IJFCS	16 (5)	2005	2017	IJFCS	30 (6–7)	2019
2005	JALC	12 (1–2)	2007	2018	TCS	798	2019
2006	TCS	387 (2)	2007	2019	IC	284	2022
2007	IJFCS	19 (4)	2008	2020	JALC	28 (1–3)	2023
2008	TCS	410 (35)	2009	2021	IJFCS	in progress	
2009	JALC	15 (1–2)	2010	2022	TCS	in progress	
2010	IJFCS	23 (1)	2012	2023	IC	planned	
2011	TCS	449	2012				

Figure 3: Journal publication of selected papers.

one can find many full versions of a certain DCFS on a fixed place and not distributed over a lot of journals. For the first editions, the full versions appeared in *Journal of Automata, Languages, and Combinatorics* (JALC), a journal edited by the University of Magdeburg with persons in the editing staff, which also were involved in the program and organizing committees. Later the journals *Theoretical Computer Science* (TCS) from Elsevier B.V., *International Journal of Foundations of Computer Science* (IJFCS) from World Scientific Publishing Co., and *Information and Computation* (IC) from Elsevier B.V. were involved. The list in Figure 3 gives the journal in relation to the year of the conference.

The development of the number of accepted papers is shown in Figure 4. It is worth mentioning that the invited contributions are not included in the statistics. In most years there were additionally 4 invited presentations and papers. However, since in the early years the spirit of DCAGRS/DCFS was that of an intense workshop, at that times the number of invited speakers was higher, with a maximum of 8 speakers in 2004. Though from the very beginning all submitted papers were peer reviewed by at least three reviewers, respectively, the PCs had to work in the classical way without the support of a more or less professional conference managing system. Due to this fact, the information about the number of submitted and, thus, the number of rejected submissions is not available before 2011. The situation changed in 2011 when EasyChair came into play.

A pleasing fact is that the number of authors and their countries of affiliations has been at a good level from the beginning. This also shows that the interest in the topic has been maintained over the years and emphasizes once more that topics of descriptional complexity have a strong continuity and are not following short-lived directions. The development is shown in Figure 5.

Further information on the DCFS series (for instance programs, contents of the proceedings, special issues etc.) can be found on the web page

`http://www.informatik.uni-giessen.de/dcfs`

3 Impressions From 25 Editions of the Conference

The contents of this section consists of personal (not scientific) impressions of the first author (who did not attend all conferences such that his reflections are limited).

Mostly, the conference took place in universities or near to the universities in the towns. The exceptions were

- 2005 Como – in a theater,

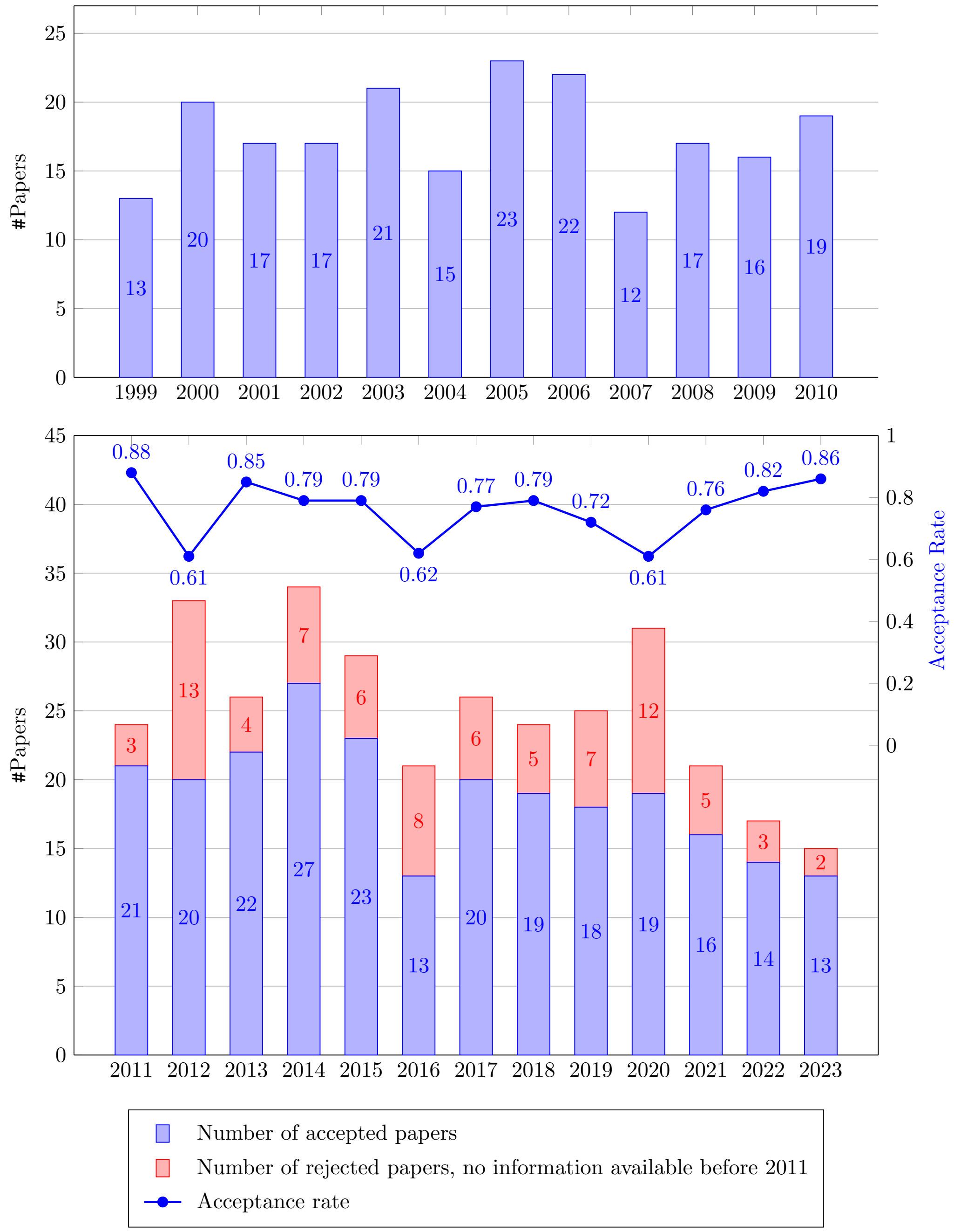


Figure 4: Development of the number of papers. Invited contributions are not included in the statistics.

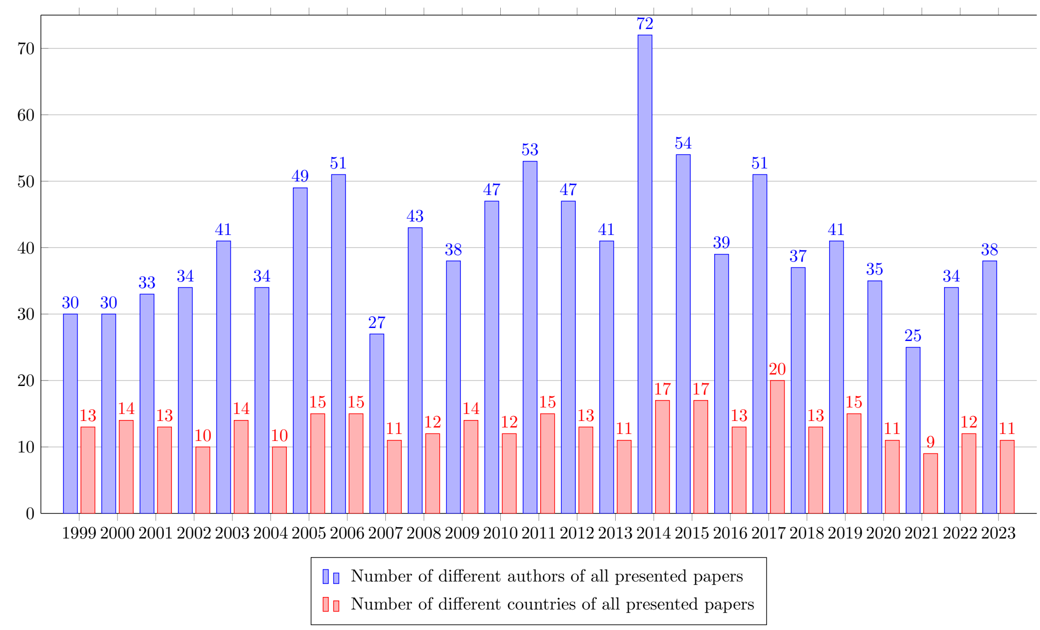


Figure 5: Development of the number of different authors and the number of different countries of all presentations. Invited contributions are included in the statistics.

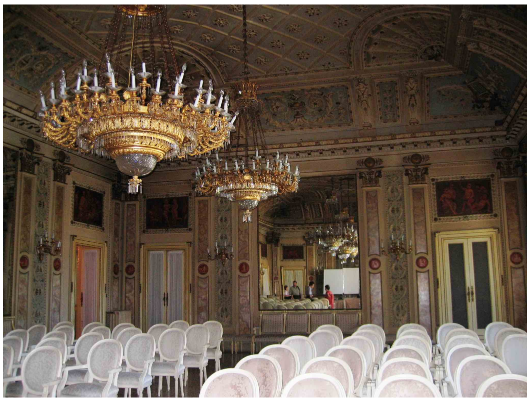
- 2007 Nový Smokovec – in a hotel in the High Tatras,
- 2011 Limburg – in a hotel,
- 2012 Braga – in a museum.

If I should give the sites which impressed me most, then two places come to my remembering:

- Sala Bianca of Teatro Sociale in Como: It was a large room with wonderful baroque design which amazed me as I entered it for the first time as well as the rest of the workshop.

- Art in the University of Saskatoon: It seemed to me that the whole university was an exhibition of different arts and science. All buildings were full of sculptures, artistic installations and other works of arts, but also some terrariums. Moreover, already on the way from the hotel to the university, I saw a lot of sculptures, etc.

In connection with conferences, workshops etc., I visited a lot of places of interest. However, I remember especially the excursions of the DCFS conferences. There are two reasons for that: I have seen many sites which also contributed to my knowledge and were not only nice places to see, and I visited landscapes of extraordinary importance. Let me mention here:



Sala Bianca, Teatro Sociale, Como

University of Saskatoon

- The sites of UNESCO World Heritage
  DCFS 2012 – visit of Guimarães, where Portugal was born,
  DCFS 2017 – excursion to Bergamo (with the wall around the upper town),
  DCFS 2018 – visit of the harbour of Lunenburg,
  DCFS 2023 – excursions to the castles and gardens of Sanssouci,
  DCFS 2010 – visit of Wanuskewin, an important place for the first nations in Canada (this is not really a UNESCO site but it belongs to the Canadian proposals)

- DCFS 2006 – excursion to the desert White Sands National Park,
  DCFS 2007 – hiking tours in the High Tatras, a UNESCO Biosphere Reservat: one tour only a short walk, one tour to the mountain hut Zamkovskeho Chata (duration appr. 2 hours), and a long tour of five hours to the Téryho Chata,
  DCFS 2008 - excursion to Green Gables, a National Historic Site of Canada (it shows places related to the book *Anne of Green Gables* by L.M. Montgomery, which tells a story on a farmer girl in the 19th century and is popular in Japan, too).

Finally, some miscellaneous reminds:

In 2002, the reception took place as a barbecue in some park near London. As we reached the place, H. Jürgensen and some of his students started to encircle it with a red net used as a fence. Since there were enough place, their handling was surprising. The reason was that it is forbidden in Ontario to take alcoholic drinks at public places. However, if you have a fence and a special permission, it is allowed. Thus they ensured wine and beer for the barbecue reception.

In 2007, the long hiking tour through the High Tatras ended at the station for a funicular. Some participants took another way of return; they used scooters.

Guimarães:
first capital of Portugal

Bergamo:
wall around the upper town

However, the way was very curvy and steep in some parts such that it was a little bit dangerous to take this way. Some of the participants arrived the base station without problems, some of them had a fall of one's scooter and a painful night.

In 2010, there were a choice in the meal of the reception, lobster or something else. Because lobster is very typical for Prince Edward Island, almost all participants chose lobster. However, almost nobody had some experience in eating lobster. Therefore anybody got a bib to protect the clothes. Then everyone did his best and mostly it was done successfully.

Helmut Jürgensen and his students
inside the fence

Markus Holzer, Martin Kutrib,
Bianca Truthe, Jürgen Dassow,
and lobsters

4 Basic Concepts of Descriptional Complexity

In order to give a deeper inside in the field of descriptional complexity, we present some of its very basics from a general abstract perspective.

We denote the set of nonnegative integers by $\mathbb{N}$. Let Σ^* denote the set of all words over a finite alphabet Σ. For the *length* of a word we write $|w|$. We use $\subseteq$ for *inclusions* and $\subset$ for *strict inclusions*. In general, the family of all languages accepted by a device of some type X is denoted by $\mathscr{L}(X)$.

In order to be general, we first formalize the intuitive notion of a representation or description of a family of languages. A *descriptional system* is a collection of encodings of items where each item *represents* or *describes* a formal language. In the following, we call the items *descriptors*, and identify the encodings of some language representation with the representation itself. More precisely, a descriptional system $\mathcal{S}$ is a set of finite descriptors such that each $D \in \mathcal{S}$ describes a formal language $L(D)$. The *family of languages represented* (or *described*) by $\mathcal{S}$ is $\mathscr{L}(\mathcal{S}) = \{ L(D) \mid D \in \mathcal{S} \}$.

A *complexity measure* for a descriptional system $\mathcal{S}$ is a total recursive mapping $c\colon \mathcal{S} \to \mathbb{N}$. From the viewpoint that a descriptional system is a collection of encoding strings, the length of the strings is a natural measure for the size. We denote it by length.

For example, nondeterministic finite automata can be encoded over some fixed alphabet. The set of these encodings is a descriptional system $\mathcal{S}$, and $\mathscr{L}(\mathcal{S})$ is the family of regular languages.

Apart from length, examples for complexity measures for nondeterministic finite automata are the *number of states* and the *number of transition*.

Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be descriptional systems with complexity measures c_1 and c_2, respectively. A total function $f\colon \mathbb{N} \to \mathbb{N}$, is said to be a *lower bound* for the increase in complexity when changing from a descriptor in $\mathcal{S}_1$ to an equivalent descriptor in $\mathcal{S}_2$, if for infinitely many $D_1 \in \mathcal{S}_1$ with $L(D_1) \in \mathscr{L}(\mathcal{S}_2)$ there exists a *minimal* $D_2 \in \mathcal{S}_2(L(D_1))$ such that $c_2(D_2) \geq f(c_1(D_1))$.

A total function $f\colon \mathbb{N} \to \mathbb{N}$ is an *upper bound* for the increase in complexity when changing from a descriptor in $\mathcal{S}_1$ to an equivalent descriptor in $\mathcal{S}_2$, if for all $D_1 \in \mathcal{S}_1$ with $L(D_1) \in \mathscr{L}(\mathcal{S}_2)$, there exists a $D_2 \in \mathcal{S}_2(L(D_1))$ such that $c_2(D_2) \leq f(c_1(D_1))$.

It may happen that the upper bound is not effectively computable. If there is no recursive upper bound, then the *trade-off* for changing from a description in $\mathcal{S}_1$ to an equivalent description in $\mathcal{S}_2$ is said to be *non-recursive*. Non-recursive trade-offs are independent of particular measures. That is, whenever the trade-off from one descriptional system to another is non-recursive, one can choose an arbitrarily large recursive function f but the gain in economy of description eventually exceeds f when changing from the former system to the latter. As an example, we consider nondeterministic pushdown automata that are used to accept regular languages. Clearly, for any such automaton there exists an equivalent finite automaton. However, the trade-off for the conversion of the pushdown automaton into the finite automaton is non-recursive.

5 Outstanding Topics

5.1 Computational Completeness with Small Resources

In the 25 edition of DCFS, there were more than 50 papers on extensions of context-free grammars (as matrix and programmed grammars etc.), insertion-deletion systems, contextual grammars, systems of grammars and automata (as Lindenmayer systems, cooperating distributed grammar systems, parallel communicating grammar systems, etc.) The problem which is mostly discussed is the following: Let a device and a numerical parameter, which describes (partly) the size of the device, be given. Let $\mathscr{L}$ be the family of languages generated by such devices. Is there a constant c such that, for each $L \in \mathscr{L}$, there is a device D which computes L and the parameter of D is at most c? Moreover, if c exists, find the minimal one.

In the sixties and seventies, a lot of variants of context-free grammars were introduced, where the sequence of the applied rules is controlled by some mechanism. We mention very informally three mechanisms and refer to [19] for details. In a *matrix grammar*, sequences of rules called matrices are given, and the generation process consists of applications of matrices, i. e., rules in the order given by the matrices; in a *programmed grammar*, with each rule, sets of successor rules are associated; in a *graph-controlled grammar*, the rules are associated to nodes of a graph and the successor rule has to be taken from the successor nodes in the graph. If one allows appearance checking, i. e., there is a set F of distinguished rules, and rules of F can be overpassed if they cannot be applied, and erasing rules, all these mentioned grammars generate all recursively enumerable languages.

As numerical parameter we take the number of nonterminals.

The first result in this direction was given by Gh. Păun in [111]. He showed that each recursively enumerable language can be generated by a matrix grammars with at most six nonterminals. An improvement was only given in 2001 in [23] and [21], where it was proved that each recursively enumerable language can be generated by a programmed grammar or a graph controlled grammar with only three nonterminals. The best known bounds where given by H. Fernau, R. Freund, M. Oswald and K. Reinhardt at DCFS'05:

Theorem 1. *(DCFS'05, [22])*

i) For each recursively enumerable language L, there is a matrix grammar with at most three nonterminals which generates L.

ii) For each recursively enumerable language L, there is a programmed grammar G with at most three nonterminals which generates L. Moreover, only two of the nonterminals are used in appearance checking mode.

iii) For each recursively enumerable language L, there is a graph controlled grammar with at most two nonterminals, both used in appearance checking mode, which generates L.

iv) The family of languages generated by graph controlled grammar with only one nonterminal used in the appearance checking mode is a proper subset of the family of all recursively enumerable languages.

We note that the results given in i) and iii) are optimal.

The operations of insertion and deletion of words are fundamental in formal language theory. They are motivated from linguistics (see contextual grammars) as well as – especially in the last 25 years – by the modeling of biological phenomena. Mostly, the insertions and deletion can only be done in a certain context, i. e., given a triple (α, w, β) of words, we can only insert w in a word x to obtain y if $x = u\alpha\beta v$ and $y = u\alpha w\beta v$ (and analogous for deletions). In context-free insertion-deletion systems, the contexts α and β are always empty. Then w can be inserted at any place in x.

A *context-free insertion-deletion system* G can be described as a 5-tuple $G = (V, T, I, D, A)$, where V and T are two alphabets with $T \subseteq V$, and $I \subseteq V^*$, $D \subseteq V^*$, and $A \subseteq V^*$ are three finite sets. The language generated by a context-free insertion-deletion system consists of all words from T^* which can be obtained from A by iterated applications of insertions of words from I and deletions of words of D.

Surprisingly, M. Margenstern, Gh. Păun, J. Rogozhin, and S. Verlan proved that already context-free insertion-deletion systems, where the length of the inserted and deleted words are short, are very powerful:

Theorem 2. *(DCFS'03, [88]) For each recursively enumerable language L, there is a context-free insertion-deletion system G where all words in I have a length at most three and all words in D have a length at most two such that G generates L.*

Two years later, S. Verlan showed that this result is optimal:

Theorem 3. *(DCFS'05, [117])*

i) A context-free insertion-deletion system, where all words of I and D have a length at most two, generates a context-free language.

ii) A context-free insertion-deletion system, where the sets I or D contain only letters, generates a context-free language.

Parallel communicating grammar systems (for short PCGSs) were introduced by Gh. Păun and L. Santean (now L. Kari) in 1989 in [112]. We only give an informal description of PCGSs. A non-returning PCGS is specified as an $(n+3)$-tuple $G = (N, K, T, G_1, G_2, \ldots, G_n)$, where V and T are alphabets, $K = \{Q_1, Q_2, \ldots, Q_n\}$ is a set of n query symbols, and, for $1 \leq i \leq n$, $G_i = (N \cup K, T, P_i, S_i)$ is a context-free grammar with an axiom $S_i \in N$. A configuration of G is an n-tuple of words over $N \cup T \cup K$. We say that $(x_1, x_2, \ldots, x_n)$ derives in one step $(y_1, y_2, \ldots, y_n)$ if and only if

a) no x_i, $1 \leq i \leq n$, contains a query symbol, and $x_i \Longrightarrow_{G_i} y_i$ for $1 \leq i \leq n$, or

b) if $x_i = z_0 Q_{i_1} z_1 Q_{i_2} z_2 \ldots Q_{i_k} z_k$ with $z_j \in (N \cup T)^*$ for $0 \leq j \leq k$ and x_{i_j} contains no query symbol for $1 \leq j \leq k$, then $y_i = z_0 x_{i_1} z_1 x_{i_2} z_2 \ldots x_{i_k} z_k$ (i. e., query symbols are replaced by the corresponding sentential form); otherwise $y_i = x_i$.

The generated language consists of all word $x_1 \in T^*$ such that there is a configuration $(x_1, x_2, \ldots, x_n)$ which can be obtained from $(S_1, S_2, \ldots, S_n)$ by some derivation steps.

The generative power of non-returning PCGSs was open for more than 10 years. In 2000, N. Mandache proved that any recursively enumerable language can be generated by some non-returning PCGS. However, the proof allows no limitation of the number of grammars (see [87]). The first bound for the number of grammars, to generate all recursively enumerable languages was presented at DCFS'05 by Gy. Vaszil, who show that eighth grammars are sufficient. An improvement was given at DCFS'09 by E. Csuhaj-Varjú and Gy. Vaszil. Hitherto, their bound is the best known one.

Theorem 4. *(DCFS'09, [18]) For any recursively enumerable language L, there is a non-returning PCGS with n grammars which generates L.*

5.2 State Complexity of Operations

One of the most studied topics at DCFS is *operational state complexity*. The topic was presented during the first DCAGRS by Sheng Yu in this invited talk *State Complexity of Regular Languages* [120]. Dozens of papers deal with various aspects of this field. Before we start our short tour through this topic we present its basic idea.

Let $\circ$ be a fixed binary operation on languages from a family $\mathscr{L}$ that is closed under the operation. Then the $\circ$-*operation problem* can be stated as follows:

- Given a language descriptor A of size m and a language descriptor B of size n such that $L(A) \in \mathscr{L}$ and $L(B) \in \mathscr{L}$.

- Which size is sufficient and necessary in the worst-case (in terms of n and m) for a language descriptor to describe the language $L(A) \circ L(B)$?

Obviously, this problem generalizes as well to $k$ary language operations like, for example, complementation. In particular, if language descriptors are considered that are finite automata whose size is measured by their number of states, the notion *operational state complexity* is used.

Operations on (non)deterministic finite automata

First observations concerning basic operation problems of DFAs can be found in [90], where tight bounds for some operations are stated without proof. In [76] the tight bound of 2^n states for the DFA reversal was obtained in connection with Boolean automata. After the dawn the research direction on DFAs was revitalized in [121]. The systematic study of nondeterministic finite automata originates in [38].

In general, a method to obtain upper bounds for some $\circ$-operation problem is to provide an *effective procedure* that constructs a finite automaton accepting the result of the operation applied to some given finite automata. The number of states of the automaton constructed is an upper bound for the problem. To show that an upper bound is tight for all input automata to the procedure, a family of minimal automata must be given such that the resulting automata achieve that bound. These families are called *witnesses*. Naturally, a witness can also be given by a family of languages.

In [8] an automaton-independent approach, called *quotient complexity*, that is based on derivatives of languages is presented, which turned out to be a very useful technique for proving upper bounds for DFA operations (cf. [10, 11, 12]).

To give an impression of basic types of results, we provide the bounds for some basic operations on DFAs and NFAs accepting infinite general and unary regular languages in Table 1.

Subregular languages

Table 1 also reflects the distinctions between general and unary regular languages that have been made from the early beginnings. At first glance the differences between general and unary languages are not that big for NFAs while it can be exponential for DFAs. However, even if nondeterminism is available, the limitation to unary languages can have a big impact to the operational state complexity.

It turned out, that for many state complexity issues of unary languages Landau's function

$$F(n) = \max\{\, \mathrm{lcm}(x_1, \ldots, x_k) \mid x_1, \ldots, x_k \geq 1 \text{ and } x_1 + \cdots + x_k = n \,\}$$

which gives the maximal order of the cyclic subgroups of the symmetric group on n elements, plays a crucial role. Here, lcm denotes the least common multiple.

	Infinite Languages			
	NFA		DFA	
	general	unary	general	unary
$\cup$	$m+n+1$	$m+n+1$	mn	mn
$\sim$	2^n	$2^{\Theta(\sqrt{n\cdot\log n})}$	n	n
$\cap$	mn	mn	mn	mn
R	$n+1$	n	2^n	n
$\cdot$	$m+n$	$m+n-1 \leq \cdot \leq m+n$	$m2^n - t2^{n-1}$	mn
$*$	$n+1$	$n+1$	$3\cdot 2^{n-2}$	$(n-1)^2+1$
$+$	n	n		
$\backslash$			2^n-1	n
$/$			n	n

Table 1: [31] NFA and DFA state complexities for operations on infinite languages, where t is the number of accepting states of the "left" automaton, \ denotes the left and / the right quotient by an arbitrary language. The tight lower bounds for union, intersection, and concatenation of unary DFAs require m and n to be relatively prime.

Since F depends on the irregular distribution of the prime numbers, we cannot expect to express $F(n)$ as a simple function of n. In [73, 74] the asymptotic growth rate

$$\lim_{n\to\infty}(\ln F(n)/\sqrt{n\cdot\ln n}) = 1$$

was determined, which implies the (for our purposes sufficient) rough estimate $F(n) \in 2^{\Theta(\sqrt{n\cdot\log n})}$. The connection with the complementation operation on unary languages represented by NFAs becomes evident from Table 1. This complementation is closely related to the unary NFA by DFA simulation, which causes also a state blow-up of order $F(n) \in 2^{\Theta(\sqrt{n\cdot\log n})}$. The proofs rely on a a normal form for unary NFAs introduced in [15]. It reads as follows.

Each n-state NFA over a unary alphabet can effectively be converted into an equivalent $O(n^2)$-state NFA consisting of an initial deterministic tail and some disjoint deterministic loops, where the automaton makes only a single nondeterministic decision after passing through the initial tail, which chooses one of the loops.

Apart from unary and finite languages, many other subregular language families have been considered from the viewpoint of operational state complexity. The systematic investigation of the descriptional complexity of such families originates in [5], where the state costs of determinizations are considered. The number of papers at DCFS dealing with this topic is quite huge and cannot be covered here. A comprehensive survey with valuable and detailed references is [24].

Universal witnesses

We mentioned above that the tightness of upper bounds is often shown by providing suitable witness languages. Interestingly, in [9] a witness over a ternary alphabet is obtained that shows the tightness of the DFA upper bounds for the operations union, intersection, concatenation, Kleene star, and reversal simultaneously. Therefore, it is called a *universal witness*. The universal witness is not always optimal with respect to the underlying alphabet size. However, from the state complexity view it can be seen as the most complex regular language. In particular, it can be used for even more. It is a witness for the maximal bounds on the number of atoms, the quotient complexity of atoms, the size of the syntactic semigroup, and about two dozen combined operations, where only a few require slightly modified versions of the universal witness. Further applications can be found in [13, 14].

Magic numbers

In connection with the well-known subset construction, a fundamental question was raised in [45]: Does there always exists a minimal n-state NFA whose equivalent minimal DFA has α states, for all n and α satisfying $n \leq \alpha \leq 2^n$. A number α not satisfying this condition is called a *magic number* (for n).

It was shown in [51] that no magic numbers exist for general regular languages over a ternary alphabet. For NFAs over a two-letter alphabet it was shown that $\alpha = 2^n - 2^k$ or $2^n - 2^k - 1$, for $0 \leq k \leq n/2 - 2$ [45], and $\alpha = 2^n - k$, for $5 \leq k \leq 2n - 2$ and some coprimality condition for k [46], are non-magic. In [53] it was proven that the integer α is non-magic, if $n \leq \alpha \leq 2^{\sqrt[3]{n}}$. Further non-magic numbers for a two-letter input alphabet were identified in [25] and [91].

Magic numbers for unary NFAs were studied in [26] by revising the Chrobak normal-form for NFAs. In the same paper also a brief historical summary of the magic number problem can be found. More general, magic numbers for several subregular language families were investigated in [37]. Further results on the magic number problem (in particular in relation to the operation problem on regular languages) can be found, for example, in [53, 50].

Further important sub-topics

As implied by the definition, so far, here we deal with the language operation problem in terms of worst-case complexity. However, the magic number problem can be seen as generalization. As opposed to the worst case, the range of state complexities that may result from an operation is considered. So, it is natural to look at the average case as well. In his invited talk *Size matters, but let's have it on average* at DCFS 2023, Rogério Reis considered various facets of this exciting field (see also [7, 98]).

Turning to another sub-topic, we recall that two words over a common alphabet are said to be *Parikh-equivalent* if and only if they are equal up to a permutation of their symbols or, equivalently, for each letter the number of its occurrences in the two words is the same. This notion extends in a natural way to languages, where two languages are Parikh-equivalent when for each word in the one language there is a Parikh-equivalent word in the other and vice versa. Inspired by the famous result of Parikh that each context-free language is Parikh-equivalent to some regular language [105], Parikh-equivalence has been connected to descriptional complexity issues. For example, in [75] the operational state complexity has been considered under parikh equivalence. That is, the resulting finite automaton must accept a language that is Parikh-equivalent to the precise language only.

Finally, the operation problems have been investigated not only for the devices DFA and NFA. A bunch of further devices have been considered. We mention only a few of them exemplarily. In her invited talk *Self-Verifying Finite Automata and Descriptional Complexity* at DCFS 2016, Galina Jirásková presented various aspects of descriptional complexity, including operation problems, on self-verifying finite automata [54]. See also [47] in this respect. The operation problems for two-way DFA were investigated in [56]. Alternating and Boolean automata are the devices considered, for example, in [43, 44, 52, 55, 76, 77]. The state complexity of operations on unambiguous finite automata and their languages is the main topic in [48].

5.3 Computational Models and Descriptional Complexity

A lot of work related to computation models, their descriptional complexity and other related properties has been done. More than 150 papers presented at DCFS investigate some kind of machines. It is impossible to briefly summarize and present in a complete way all the results obtained in this context. We just give some relevant examples and pointers to the literature.

While studying a computational model, the first question concerns its computational power. In the case of devices recognizing languages, this leads to investigate the class of accepted languages. The second natural question concerns the succinctness. In particular, when a computational model can be simulated by another one, it is quite natural to investigate the cost of such a simulation in terms of the size of the descriptions. In the Introduction we already mentioned the classical example that can used to introduce descriptional complexity, namely the simulation of NFAs automata by DFAs, given by the subset construction: each one-way nondeterministic automaton with n states can be simulated by an equivalent deterministic finite automaton with 2^n states. Furthermore, it is well-known that in the worst case such a cost cannot be reduced [95].

During the DCFS conferences a lot of results have been presented concerning the costs of the relative succinctness of computational models.

Finite automata

We just mentioned the exponential cost of the simulation of NFAs by DFAs. In a paper presented at DCAGRS 1999, M. Kappes proved this cost cannot be reduced even when simulating *deterministic finite automata with multiple initial states*, i.e., if the only nondeterministic choice can be taken at the beginning of the computation, to choose the initial state in a given set [59]. This research was refined in [42] by giving an exact bound that keeps into account the cardinality of the set of possible initial states, besides the cardinality of the set of states of the simulated automaton.

Even for many non-trivial *subclasses of regular languages* (e.g., star-free languages, strictly locally testable languages) the cost of the elimination on nondeterminism remains exponential [5].

In the above mentioned results, the focus is on *one-way finite automata*, i.e., automata that scans the input tape from left to right. It is well-known that the computational power does not increase if the head can be moved in both directions, so obtaining *two-way finite automata*. A long-standing open problem related to these devices is the cost of the elimination of nondeterminism using two-way motion. This problem was formulated in 1978 by Sakoda and Sipser, who asked the costs, in terms of states, of the conversions of one-way and two-way NFAs into equivalent *two-way DFAs*, and it is still open [114]. For both questions the best-known upper bounds are exponential, while the lower bounds are polynomial. Several contributions related to this problem and, more in general, to two-way finite automata have been presented in DCFS conferences. We point out just few of them.

In his invited lecture at DCFS 2012, Ch. Kapoutsis presented *Minicomplexity* [58], a complexity theory for two-way automata which brings together several seemingly detached concepts and results.

V. Geffert and L. Isonová presented a translation of the Sakoda and Sipser question on two-way automata, to an analogous question on *pebble automata* [27].

The maximum length of the shortest string accepted by an n-state two-way finite automaton is known to be exponential in n. However, its exact value is not yet known. Recent contributions towards the solution of this problem have been presented at DCFS 2020 and 2023 [64, 89].

Extension of two-way automata, with restricted rewriting capabilities, have been also considered and they will be mentioned later.

Pushdown automata

As we just discussed, a lot of contributions related to regular languages and finite automata have been given. To represent a regular language, we could use a device from a more powerful class of machines. For instance, we could use a pushdown automaton. So the question of comparing the relative succinctness of finite automata and equivalent pushdown automata arises. This question was solved longtime ago by Meyer and Fischer by proving nonrecursive trade-offs [95].

One could ask what happens if pushdown automata in some restricted form are considered. One possible restriction is to require that the height of the pushdown store is bounded by a constant. This leads to the definition of *constant height pushdown automata*, introduced and firstly studied by Z. Bednárová, V. Geffert, C. Mereghetti, and B. Palano [29, 3]. Under this restriction, the trade-off to finite automata is recursive. In particular, the size cost of the conversion of nondeterministic constant height pushdown automata into equivalent one-way deterministic finite automata is double exponential. So, constant height pushdown automata are very interesting for their succinctness. This line of research has been recently deepened. It is well-known that it cannot be decided whether the language accepted by a pushdown automaton is regular [95]. Notice that there exists pushdown automata that accept regular languages using a non-constant amount of pushdown store. This leads to the different question of deciding for a pushdown automaton whether there exists a constant h such that each string in the accepted language has an accepting computation using height at most h. In [110] it has been proved that also this property is undecidable. It remains undecidable when the pushdown alphabet is unary, i.e, when the machine is a one-counter automaton [109].

More in general, computations of pushdown automata have been analyzed in [6], introducing and studying *pushdown information* (roughly the properties of strings written on the pushdown store during computations of stateless pushdown automata), and in [28, 85], where the descriptional complexity of the *pushdown store language*, i.e., the set of strings that appears of the pushdown during an accepting computation, is studied. We point out that this language is regular.

A. Malcher considered *finite-turn pushdown automata*, namely pushdown automata that can switch from push to pop operations a number of time bounded by a fixed constant k, proving a series of interesting non-recursive trade-offs (e.g., from k-turns to $k + 1$ turns, for each $k \geq 1$) [84].

While in the case of finite automata, having a two-way input tape does not increase the computational power, in the case of pushdown automata the situation is different. In fact, *two-way pushdown automata* can recognize even non-context-free languages. At the moment is still unknown if these devices are able to recognize all context-sensitive languages. In [86] the authors started an investigation on these devices in the case of a constant number of head reversals.

Questions related to *input driven pushdown automata* (also known as *nested word* or *visibly pushdown automata*) have been investigated in [106, 100, 36].

In [40] the authors consider *one-time nondeterministic* finite and pushdown automata. In these devices, whenever a guess is performed, it remains fixed for the rest of the computation. In the case of finite automata, the state increase to equivalent deterministic devices is bounded by an exponential function, while in the case of pushdown automata nonrecursive size trade-offs have been proved.

Turing machines and their variants, Cellular automata

The model of *restarting automata* has been the subject of many papers presented at DCFS and in other related conferences. This is a formal model for the *analysis by reduction*, which is used in linguistic to analyze sentences of natural languages. Roughly, this technique consists in a stepwise simplification of a given sentence in such a way that the syntactical correctness or incorrectness of the sentence is not affected. Such a process can be modeled by machines having a flexible tape where, at some point, such a simplification is performed and then the computation is restarted.

In his invited lecture *On Restarting Automata with Window Size One* at DCFS 2011, F. Otto presented a general overview of the most important variants of restarting automata, together with new results on restarting automata with window size one [101]. Several other contributions in this area have been presented in DCFS conferences. Among them, we address the reader to [41, 72, 102, 103].

Turing machines with restricted rewriting capabilities have been considered in several papers. At DCFS 2005, B. Durak presented *worm automata*. These devices are two-way finite automata equipped with a write-once track. In spite of this possibility, they still recognize only regular languages [20].

More recently, several results on *limited automata* have been presented. These devices are single-tape Turing machines in which the content of each tape cell can be rewritten only in the first d visits, for a fixed integer $d \geq 0$. In case $d \leq 1$

these devices accept only regular languages, while for each fixed $d > 1$, they characterize the class of context-free languages. The conversion from nondeterministic 1-limited automata into equivalent one-way deterministic finite automata costs, in the worst case, double exponential in size. So these devices can be extremely succinct [108]. Other results on the descriptional complexity of limited automata have presented in [71, 34]. A survey on limited automata has been given in the invited lecture *Limited Automata: Properties, Complexity and Variants* by G. Pighizzini at DCFS 2019 [107].

In his invited lecture *The Descriptional Power of Sublogarithmic Resource Bounded Turing Machines* at DCFS 2007, C. Mereghetti presented a complete picture of lower bounds on space and input head reversals for deterministic, nondeterministic, and alternating Turing machines accepting nonregular languages [92].

In [66], M. Kutrib investigated multitape Turing machines having a restricted number of nondeterministic steps, proving the existence of a nondeterministic language hierarchy between real time and linear time.

Among other computational models, it is also suitable to mention *cellular automata*. They have been the subject of invited lectures *Cellular Automata and Descriptional Complexity* by A. Malcher at DCFS 2006 [83] and *Linear Algebra Based Bounds for One-Dimensional Cellular Automat* by J. Kari at DCFS 2011 [60]. The descriptional complexity and the properties of several variants of cellular automata (e.g., one-way and two-way) have been the subject of various talks (e.g [80, 81, 82, 68]).

Non classical computation modes: probabilistic and quantum

Besides classical modes of computations, mainly based on determinism, nondeterminism, and alternation, several contributions of other modes have been presented.

At DCFS 2009, Ch. Baier gave the invited talk *Probabilistic Automata over Infinite Words: Expressiveness, Efficiency, and Decidability* [2]. Probabilistic models have been also considered in [96, 49].

Quantum automata and quantum computations have been the subject of invited talks *Descriptional complexity issues in quantum computing* and *Succinctness in quantum information processing* by J. Gruska at DCAGRS 1999 and DCFS 2003 [32, 33], *Some formal tools for analyzing quantum automata* by A. Bertoni at DCFS 2005 [4], and *Recent Developments in Quantum Algorithms and Complexity* A. Ambainis at DCFS 2014 [1]. Several contributions to this area have been presented [93, 118, 119]. In [94] the authors compare the succinctness of deterministic, nondeterministic, probabilistic and quantum finite automata.

Non-recursive trade-offs

A general survey on non-recursive trade-offs, with a unifying approach to the proof of them, has been given by M. Kutrib in his invited talk *The phenomenon of non-recursive trade-offs* at DCFS 2004 [67]. Further developments on this phenomenon have been obtained in [30].

In [57], Ch. Kapoutsis presented non-recursive trade-offs for multi-head two-way finite automata and multi-counter automata. We already mentioned the paper by A. Malcher with non-recursive trade-offs for related to finite-turn pushdown automata [84]. In many other papers (e.g. related to pushdown automata [86, 40, 109]) results presenting non-recursive trade-offs have been given.

Ambiguity and measures of nondeterminism

In this invited talk *Descriptional complexity of nfa of different ambiguity* at DCFS 2004, H. Leung presented relationships between descriptional complexity and ambiguity degree for nondeterministic finite automata [78]. Further results on this topic are given in [65]. The case of Büchi automata was considered in [99],

Unambiguity in automata theory was the title of the invited lecture given by Th. Colcombet at DCFS 2015 [17]: the concept of unambiguity, seen as a generalization of determinism, has been explored not only in automata on finite words, but in some extensions of them as, e.g., automata on infinite trees and tropical automata.

In [104] various *measure of nondeterminism* for finite automata have been investigated and compared. This idea was further explored in some subsequent papers. Among them we mention [61, 63]. More recently, also measures for *alternating automata* have been introduced and studied [62, 35]

References

[1] Ambainis, A.: Recent developments in quantum algorithms and complexity. In: Descriptional Complexity of Formal Systems (DCFS 2014). LNCS, vol. 8614, pp. 1–4. Springer (2014). https://doi.org/10.1007/978-3-319-09704-6_1

[2] Baier, C., Bertrand, N., Größer, M.: Probabilistic automata over infinite words: Expressiveness, efficiency, and decidability. In: Descriptional Complexity of Formal Systems (DCFS 2009). EPTCS, vol. 3, pp. 3–16 (2009). https://doi.org/10.4204/EPTCS.3.1

[3] Bednárová, Z., Geffert, V., Mereghetti, C., Palano, B.: Removing nondeterminism in constant height pushdown automata. Inf. Comput. **237**, 257–267 (2014). https://doi.org/10.1016/j.ic.2014.03.002

[4] Bertoni, A., Mereghetti, C., Palano, B.: Some formal tools for analyzing quantum automata. Theor. Comput. Sci. **356**, 14–25 (2006). https://doi.org/10.1016/j.tcs.2006.01.042

[5] Bordihn, H., Holzer, M., Kutrib, M.: Determinization of finite automata accepting subregular languages. Theor. Comput. Sci. **410**, 3209–3222 (2009). https://doi.org/10.1016/j.tcs.2009.05.019

[6] Bordihn, H., Jürgensen, H.: Pushdown information. In: Descriptional Complexity of Formal Systems (DCFS 2004). pp. 111–120. Report No. 619, Department of Computer Science, The University of Western Ontario, Canada (2004)

[7] Broda, S., Machiavelo, A., Moreira, N., Reis, R.: Analytic combinatorics and descriptional complexity of regular languages on average. SIGACT News **51**, 38–56 (2020). https://doi.org/10.1145/3388392.3388401

[8] Brzozowski, J.A.: Quotient complexity of regular languages. J. Autom. Lang. Comb. **15**, 71–89 (2010). https://doi.org/10.25596/jalc-2010-071

[9] Brzozowski, J.A.: In search of the most complex regular language. Int. J. Found. Comput. Sci. **24**, 692–708 (2013). https://doi.org/10.1142/S0129054113400133

[10] Brzozowski, J.A., Jirásková, G., Li, B.: Quotient complexity of ideal languages. Theor. Comput. Sci. **470**, 36–52 (2013). https://doi.org/10.1016/j.tcs.2012.10.055

[11] Brzozowski, J.A., Jirásková, G., Zou, C.: Quotient complexity of closed languages. Theor. Comput. Sci. **54**, 277–292 (2014). https://doi.org/10.1007/s00224-013-9515-7

[12] Brzozowski, J.A., Liu, B.: Quotient complexity of star-free languages. Int. J. Found. Comput. Sci. **23**, 1261–1276 (2012). https://doi.org/10.1142/S0129054112400515

[13] Brzozowski, J.A., Liu, D.: Universal witnesses for state complexity of basic operations combined with reversal. In: Implementation and Application of Automata (CIAA 2013). LNCS, vol. 7982, pp. 72–83. Springer (2013). https://doi.org/10.1007/978-3-642-39274-0_8

[14] Brzozowski, J.A., Liu, D.: Universal witnesses for state complexity of Boolean operations and concatenation combined with star. In: Descriptional Complexity of Formal Systems (DCFS 2013). LNCS, vol. 8031, pp. 30–41. Springer (2013). https://doi.org/10.1007/978-3-642-39310-5_5

[15] Chrobak, M.: Finite automata and unary languages. Theor. Comput. Sci. **47**, 149–158 (1986). https://doi.org/10.1016/0304-3975(86)90142-8, errata: [16]

[16] Chrobak, M.: Errata to "finite automata and unary languages". Theor. Comput. Sci. **302**, 497–498 (2003). https://doi.org/10.1016/S0304-3975(03)00136-1

[17] Colcombet, T.: Unambiguity in automata theory. In: Descriptional Complexity of Formal Systems (DCFS 2015). LNCS, vol. 9118, pp. 3–18. Springer (2015). https://doi.org/10.1007/978-3-319-19225-3_1

[18] Csuhaj-Varjú, E., Vaszil, G.: On the descriptional complexity of context-free non-returning pc grammar systems. J. Autom. Lang. Comb. **15**, 91–105 (2010). https://doi.org/10.25596/jalc-2010-091

[19] Dassow, J., Păun, G.: Regulated Rewriting in Formal Language Theory. Springer (1989)

[20] Durak, B.: Two-way finite automata with a write-once track. J. Autom. Lang. Comb. **12**(1-2), 97–115 (2007). https://doi.org/10.25596/jalc-2007-097

[21] Fernau, H.: Nonterminal complexity of programmed grammars. In: Machines, Computations, and Universality (MCU 2001). LNCS, vol. 2055, pp. 202–213. Springer (2001). https://doi.org/10.1007/3-540-45132-3_13

[22] Fernau, H., Freund, R., Oswald, M., Reinhardt, K.: Refining the nonterminal complexity of graph-controlled, programmed, and matrix grammars. J. Autom. Lang. Comb. **12**, 117–138 (2007). https://doi.org/10.25596/jalc-2007-117

[23] Freund, R., Păun, G.: On the number of nonterminal symbols in graph-controlled, programmed, and matrix grammars. In: Machines, Computations, and Universality (MCU 2001). LNCS, vol. 2055, pp. 214–225. Springer (2001). https://doi.org/10.1007/3-540-45132-3_14

[24] Gao, Y., Moreira, N., Reis, R., Yu, S.: A survey on operational state complexity. J. Autom. Lang. Comb. **21**, 251–310 (2016). https://doi.org/10.25596/jalc-2016-251

[25] Geffert, V.: (Non)determinism and the size of one-way finite automata. In: Descriptional Complexity of Formal Systems (DCFS 2005). pp. 23–37. Rapporto Tecnico 06-05, Università degli Studi di Milano (2005)

[26] Geffert, V.: Magic numbers in the state hierarchy of finite automata. Inform. Comput. **205**, 1652–1670 (2007). https://doi.org/10.1016/j.ic.2007.07.001

[27] Geffert, V., Istonová, L.: Translation from classical two-way automata to pebble two-way automata. RAIRO Theor. Informatics Appl. **44**, 507–523 (2010). https://doi.org/10.1051/ita/2011001

[28] Geffert, V., Malcher, A., Meckel, K., Mereghetti, C., Palano, B.: A direct construction of finite state automata for pushdown store languages. In: Descriptional Complexity of Formal Systems (DCFS 2013). LNCS, vol. 8031, pp. 90–101. Springer (2013). https://doi.org/10.1007/978-3-642-39310-5_10

[29] Geffert, V., Mereghetti, C., Palano, B.: More concise representation of regular languages by automata and regular expressions. Inf. Comput. **208**, 385–394 (2010). https://doi.org/10.1016/j.ic.2010.01.002

[30] Gruber, H., Holzer, M., Kutrib, M.: On measuring non-recursive trade-offs. J. Autom. Lang. Comb. **15**, 107–120 (2010). https://doi.org/10.25596/jalc-2010-107

[31] Gruber, H., Holzer, M., Kutrib, M.: Descriptional complexity of regular languages. In: Handbook of Automata Theory, pp. 411–457. European Mathematical Society Publishing House (2021). https://doi.org/10.4171/Automata-1/12

[32] Gruska, J.: Descriptional complexity issues in quantum computing. J. Autom. Lang. Comb. **5**, 191–218 (2000). https://doi.org/10.25596/jalc-2000-191

[33] Gruska, J.: Succinctness in quantum information processing. In: Descriptional Complexity of Formal Systems (DCFS 2003). pp. 15–25. MTA SZTAKI, Hungarian Academy of Sciences (2003)

[34] Guillon, B., Prigioniero, L.: Linear-time limited automata. Theor. Comput. Sci. **798**, 95–108 (2019). https://doi.org/10.1016/j.tcs.2019.03.037

[35] Han, Y., Kim, S., Ko, S., Salomaa, K.: Existential and universal width of alternating finite automata. In: Descriptional Complexity of Formal Systems (DCFS 2023). LNCS, vol. 13918, pp. 51–64. Springer (2023). https://doi.org/10.1007/978-3-031-34326-1_4

[36] Han, Y., Ko, S., Salomaa, K.: Limited nondeterminism of input-driven pushdown automata: Decidability and complexity. In: Descriptional Complexity of Formal Systems (DCFS 2019). LNCS, vol. 11612, pp. 158–170. Springer (2019). https://doi.org/10.1007/978-3-030-23247-4_12

[37] Holzer, M., Jakobi, S., Kutrib, M.: The magic number problem for subregular language families. Int. J. Found. Comput. Sci. **23**, 115–131 (2012). https://doi.org/10.1142/S0129054112400084

[38] Holzer, M., Kutrib, M.: Nondeterministic descriptional complexity of regular languages. Int. J. Found. Comput. Sci. **14**, 1087–1102 (2003). https://doi.org/10.1142/S0129054103002199

[39] Holzer, M., Kutrib, M.: Descriptional complexity – An introductory survey. In: Scientific Applications of Language Methods, pp. 1–58. Imperial College Press (2010)

[40] Holzer, M., Kutrib, M.: One-time nondeterministic computations. Int. J. Found. Comput. Sci. **30**, 1069–1089 (2019). https://doi.org/10.1142/S012905411940029X

[41] Holzer, M., Kutrib, M., Reimann, J.: Non-recursive trade-offs for deterministic restarting automata. J. Autom. Lang. Comb. **12**, 195–213 (2007). https://doi.org/10.25596/jalc-2007-195

[42] Holzer, M., Salomaa, K., Yu, S.: On the state complexity of k-entry deterministic finite automata. J. Autom. Lang. Comb. **6**, 453–466 (2001). https://doi.org/10.25596/jalc-2001-453

[43] Hospodár, M., Jirásková, G.: The complexity of concatenation on deterministic and alternating finite automata. RAIRO Inform. Théor. **52**, 153–168 (2018). https://doi.org/10.1051/ita/2018011

[44] Hospodár, M., Jirásková, G., Krajnáková, I.: Operations on Boolean and alternating finite automata. In: Computer Science Symposium in Russia (CSR 2018). LNCS, vol. 10846, pp. 181–193. Springer (2018). https://doi.org/10.1007/978-3-319-90530-3_16

[45] Iwama, K., Kambayashi, Y., Takaki, K.: Tight bounds on the number of states of DFAs that are equivalent to n-state NFAs. Theor. Comput. Sci. **237**, 485–494 (2000). https://doi.org/10.1016/S0304-3975(00)00029-3

[46] Iwama, K., Matsuura, A., Paterson, M.: A family of NFAs which need $2^n - \alpha$ deterministic states. Theor. Comput. Sci. **301**, 451–462 (2003). https://doi.org/10.1016/S0304-3975(02)00891-5

[47] Jirásek, J.S., Jirásková, G., Szabari, A.: Operations on self-verifying finite automata. In: Computer Science Symposium in Russia (CSR 2015). LNCS, vol. 9139, pp. 231–261. Springer (2015). https://doi.org/10.1007/978-3-319-20297-6_16

[48] Jirásek Jr., J., Jirásková, G., Sebej, J.: Operations on unambiguous finite automata. Int. J. Found. Comput. Sci. **29**, 861–876 (2018). https://doi.org/10.1142/S012905411842008X

[49] Jirásková, G.: Note on the complexity of Las Vegas automata problems. RAIRO Theor. Informatics Appl. **40**, 501–510 (2006). https://doi.org/10.1051/ita:2006033

[50] Jirásková, G.: Concatenation of regular languages and descriptional complexity. Theor. Comput. Sci. **49**, 306–318 (2011). https://doi.org/10.1007/s00224-011-9318-7

[51] Jirásková, G.: Magic numbers and ternary alphabet. Int. J. Found. Comput. Sci. **22**, 331–344 (2011). https://doi.org/10.1142/S0129054111008076

[52] Jirásková, G.: Descriptional complexity of operations on alternating and Boolean automata. In: Computer Science Symposium in Russia (CSR 2012). LNCS, vol. 7353, pp. 196–204. Springer (2012). https://doi.org/10.1007/978-3-642-30642-6_19

[53] Jirásková, G.: The ranges of state complexities for complement, star, and reversal of regular languages. Int. J. Found. Comput. Sci. **25**, 101 (2014). https://doi.org/10.1142/S0129054114500063

[54] Jirásková, G.: Self-verifying finite automata and descriptional complexity. In: Descriptional Complexity of Formal Systems (DCFS 2016). LNCS, vol. 9777, pp. 29–44. Springer (2016). https://doi.org/10.1007/978-3-319-41114-9_3

[55] Jirásková, G., Krajnáková, I.: Square on deterministic, alternating, and Boolean finite automata. Int. J. Found. Comput. Sci. **30**, 1117–1134 (2019). https://doi.org/10.1142/S0129054119400318

[56] Jirásková, G., Okhotin, A.: On the state complexity of operations on two-way finite automata. In: Developments in Language Theory (DLT 2008). LNCS, vol. 5257, pp. 443–454. Springer (2008). https://doi.org/10.1007/978-3-540-85780-8_35

[57] Kapoutsis, C.A.: Non-recursive trade-offs for two-way machines. Int. J. Found. Comput. Sci. **16**, 943–956 (2005). https://doi.org/10.1142/S012905410500339X

[58] Kapoutsis, C.A.: Minicomplexity. J. Autom. Lang. Comb. **17**, 205–224 (2012). https://doi.org/10.25596/jalc-2012-205

[59] Kappes, M.: Descriptional complexity of deterministic finite automata with multiple initial states. J. Autom. Lang. Comb. **5**, 269–278 (2000). https://doi.org/10.25596/jalc-2000-269

[60] Kari, J.: Linear algebra based bounds for one-dimensional cellular automata. In: Descriptional Complexity of Formal Systems (DCFS 2011). LNCS, vol. 6808, pp. 1–7. Springer (2011). https://doi.org/10.1007/978-3-642-22600-7_1

[61] Keeler, C., Salomaa, K.: Branching measures and nearly acyclic NFAs. Int. J. Found. Comput. Sci. **30**, 1135–1155 (2019). https://doi.org/10.1142/S012905411940032X

[62] Keeler, C., Salomaa, K.: Width measures of alternating finite automata. In: Descriptional Complexity of Formal Systems (DCFS 2021). LNCS, vol. 13037, pp. 88–99. Springer (2021). https://doi.org/10.1007/978-3-030-93489-7_8

[63] Keeler, C., Salomaa, K.: Structural properties of NFAs and growth rates of nondeterminism measures. Inf. Comput. **284**, 104690 (2022). https://doi.org/10.1016/j.ic.2021.104690

[64] Krymski, S., Okhotin, A.: Longer shortest strings in two-way finite automata. In: Descriptional Complexity of Formal Systems (DCFS 2020). LNCS, vol. 12442, pp. 104–116. Springer (2020). https://doi.org/10.1007/978-3-030-62536-8_9

[65] Kupke, J.: A powerful tool in lower-bounding constantly ambiguous automata. In: Descriptional Complexity of Formal Systems (DCFS 2006). pp. 276–284. New Mexico State University, Las Cruces, New Mexico, USA (2006)

[66] Kutrib, M.: Refining nondeterminism below linear time. J. Autom. Lang. Comb. **7**, 533–547 (2002). https://doi.org/10.25596/jalc-2002-533

[67] Kutrib, M.: The phenomenon of non-recursive trade-offs. Int. J. Found. Comput. Sci. **16**, 957–973 (2005). https://doi.org/10.1142/S0129054105003406

[68] Kutrib, M., Malcher, A.: One-way cellular automata, bounded languages, and minimal communication. J. Autom. Lang. Comb. **15**, 135–153 (2010). https://doi.org/10.25596/jalc-2010-135

[69] Kutrib, M., Moreira, N., Pighizzini, G., Reis, R.: Hot current topics of descriptional complexity. In: Advancing Research in Information and Communication Technology – IFIP's Exciting First 60+ Years, IFIP Advances in Information and Communication Technology, vol. 600, pp. 3–28. Springer (2021). https://doi.org/10.1007/978-3-030-81701-5_1

[70] Kutrib, M., Pighizzini, G.: Recent trends in descriptional complexity of formal languages. Bull. EATCS **111** (2013)

[71] Kutrib, M., Pighizzini, G., Wendlandt, M.: Descriptional complexity of limited automata. Inf. Comput. **259**, 259–276 (2018). https://doi.org/10.1016/j.ic.2017.09.005

[72] Kutrib, M., Reimann, J.: Optimal simulations of weak restarting automata. Int. J. Found. Comput. Sci. **19**, 795–811 (2008). https://doi.org/10.1142/S0129054108005966

[73] Landau, E.: Über die Maximalordnung der Permutationen gegebenen Grades. Archiv der Math. und Phys. **3**, 92–103 (1903)

[74] Landau, E.: Handbuch der Lehre von der Verteilung der Primzahlen. Teubner, Leipzig (1909)

[75] Lavado, G.J., Pighizzini, G., Seki, S.: Operational state complexity under Parikh equivalence. In: Descriptional Complexity of Formal Systems (DCFS 2014). LNCS, vol. 8614, pp. 294–305. Springer (2014). https://doi.org/10.1007/978-3-319-09704-6_26

[76] Leiss, E.L.: Succinct representation of regular languages by Boolean automata. Theor. Comput. Sci. **13**, 323–330 (1981). https://doi.org/10.1016/S0304-3975(81)80005-9

[77] Leiss, E.L.: Succinct representation of regular languages by Boolean automata. II. Theor. Comput. Sci. **38**, 133–136 (1985). https://doi.org/10.1016/0304-3975(85)90215-4

[78] Leung, H.: Descriptional complexity of nfa of different ambiguity. Int. J. Found. Comput. Sci. **16**, 975–984 (2005). https://doi.org/10.1142/S0129054105003418

[79] Lupanov, O.B.: A comparison of two types of finite sources. Problemy Kybernetiki **9**, 321–326 (1963), (in Russian), German translation: Über den Vergleich zweier Typen endlicher Quellen. Probleme der Kybernetik 6 (1966), 328–335

[80] Malcher, A.: Descriptional complexity of cellular automata and decidability questions. J. Autom. Lang. Comb. **7**, 549–560 (2002). https://doi.org/10.25596/jalc-2002-549

[81] Malcher, A.: On one-way cellular automata with a fixed number of cells. Fundam. Inform. **58**, 355–368 (2003)

[82] Malcher, A.: On two-way communication in cellular automata with a fixed number of cells. Theor. Comput. Sci. **330**, 325–338 (2005). https://doi.org/10.1016/j.tcs.2004.04.014

[83] Malcher, A.: Cellular automata and descriptional complexity. In: Descriptional Complexity of Formal Systems (DCFS 2006). pp. 26–40. New Mexico State University, Las Cruces, New Mexico, USA (2006)

[84] Malcher, A.: On recursive and non-recursive trade-offs between finite-turn pushdown automata. J. Autom. Lang. Comb. **12**, 265–277 (2007). https://doi.org/10.25596/jalc-2007-265

[85] Malcher, A., Meckel, K., Mereghetti, C., Palano, B.: Descriptional complexity of pushdown store languages. J. Autom. Lang. Comb. **17**, 225–244 (2012). https://doi.org/10.25596/jalc-2012-225

[86] Malcher, A., Mereghetti, C., Palano, B.: Descriptional complexity of two-way pushdown automata with restricted head reversals. Theor. Comput. Sci. **449**, 119–133 (2012). https://doi.org/10.1016/j.tcs.2012.04.007

[87] Mandache, N.: On the computational power of context-free pc grammar systems. Theor. Comput. Sci. **237**, 135–148 (2000). https://doi.org/10.1016/S0304-3975(98)00159-5

[88] Margenstern, M., Păun, G., Rogozhin, Y., Verlan, S.: Context-free insertion-deletion systems. Theor. Comput. Sci. **330**, 317–328 (2005). https://doi.org/10.1016/j.tcs.2004.06.031

[89] Martynova, O., Okhotin, A.: Shortest accepted strings for two-way finite automata: Approaching the 2^n lower bound. In: Descriptional Complexity of Formal Systems (DCFS 2023). LNCS, vol. 13918, pp. 134–145. Springer (2023). https://doi.org/10.1007/978-3-031-34326-1_10

[90] Maslov, A.N.: Estimates of the number of states of finite automata. Soviet Math. Dokl. **11**, 1373–1375 (1970), (English translation), in Russian: Dokl. Akad. Nauk SSSR 194 (1970), 1266–1268

[91] Matsuura, A., Saito, Y.: Equivalent transformation of minimal finite automata over a two-letter alphabet. In: Descriptional Complexity of Formal Systems (DCFS 2008). pp. 224–232. University of Prince Edward Island, Canada (2008)

[92] Mereghetti, C.: Testing the descriptional power of small Turing machines on non-regular language acceptance. Int. J. Found. Comput. Sci. **19**, 827–843 (2008). https://doi.org/10.1142/S012905410800598X

[93] Mereghetti, C., Palano, B.: Quantum automata for some multiperiodic languages. Theor. Comput. Sci. **387**, 177–186 (2007). https://doi.org/10.1016/j.tcs.2007.07.037

[94] Mereghetti, C., Palano, B., Pighizzini, G.: Note on the succinctness of deterministic, nondeterministic, probabilistic and quantum finite automata. RAIRO Theor. Informatics Appl. **35**, 477–490 (2001). https://doi.org/10.1051/ita:2001106

[95] Meyer, A.R., Fischer, M.J.: Economy of description by automata, grammars, and formal systems. In: Symposium on Switching and Automata Theory (SWAT 1971). pp. 188–191. IEEE (1971). https://doi.org/10.1109/SWAT.1971.11

[96] Milani, M., Pighizzini, G.: Tight bounds on the simulation of unary probabilistic automata by deterministic automata. J. Autom. Lang. Comb. **6**, 481–492 (2001). https://doi.org/10.25596/jalc-2001-481

[97] Moore, F.R.: On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata. IEEE Trans. Comput. **20**, 1211–1214 (1971). https://doi.org/10.1109/T-C.1971.223108

[98] Nicaud, C.: Average state complexity of operations on unary automata. In: Mathematical Foundations of Computer Science (MFCS 1999). LNCS, vol. 1672, pp. 231–240. Springer (1999). https://doi.org/10.1007/3-540-48340-3_21

[99] Nießner, F.: Büchi automata and their degrees of nondeterminism and ambiguity. J. Autom. Lang. Comb. **9**, 347–363 (2004). https://doi.org/10.25596/jalc-2004-347

[100] Okhotin, A., Salomaa, K.: Further closure properties of input-driven pushdown automata. Theor. Comput. Sci. **798**, 65–77 (2019). https://doi.org/10.1016/j.tcs.2019.04.006

[101] Otto, F.: On restarting automata with window size one. In: Descriptional Complexity of Formal Systems (DCFS 2011). LNCS, vol. 6808, pp. 8–33. Springer (2011). https://doi.org/10.1007/978-3-642-22600-7_2

[102] Otto, F.: On the descriptional complexity of deterministic ordered restarting automata. In: Descriptional Complexity of Formal Systems (DCFS 2014). LNCS, vol. 8614, pp. 318–329. Springer (2014). https://doi.org/10.1007/978-3-319-09704-6_28

[103] Otto, F., Kwee, K.: On the descriptional complexity of stateless deterministic ordered restarting automata. Inf. Comput. **259**, 277–302 (2018). https://doi.org/10.1016/j.ic.2017.09.006

[104] Palioudakis, A., Salomaa, K., Akl, S.G.: Comparisons between measures of nondeterminism on finite automata. In: Descriptional Complexity of Formal Systems (DCFS 2013). LNCS, vol. 8031, pp. 217–228. Springer (2013). https://doi.org/10.1007/978-3-642-39310-5_21

[105] Parikh, R.J.: On context-free languages. J. ACM **13**, 570–581 (1966). https://doi.org/10.1145/321356.321364

[106] Piao, X., Salomaa, K.: Operational state complexity of nested word automata. Theor. Comput. Sci. **410**, 3290–3302 (2009). https://doi.org/10.1016/j.tcs.2009.05.002

[107] Pighizzini, G.: Limited automata: Properties, complexity and variants. In: Descriptional Complexity of Formal Systems (DCFS 2019). LNCS, vol. 11612, pp. 57–73. Springer (2019). https://doi.org/10.1007/978-3-030-23247-4_4

[108] Pighizzini, G., Pisoni, A.: Limited automata and regular languages. Int. J. Found. Comput. Sci. **25**, 897–916 (2014). https://doi.org/10.1142/S0129054114400140

[109] Pighizzini, G., Prigioniero, L.: Pushdown and one-counter automata: Constant and non-constant memory usage. In: Descriptional Complexity of Formal Systems (DCFS 2023). LNCS, vol. 13918, pp. 146–157. Springer (2023). https://doi.org/10.1007/978-3-031-34326-1_11

[110] Pighizzini, G., Prigioniero, L.: Pushdown automata and constant height: decidability and bounds. Acta Informatica **60**, 123–144 (2023). https://doi.org/10.1007/s00236-022-00434-0

[111] Păun, G.: Six nonterminals are enough for generating each r.e. language by a matrix grammar. Internat. J. Comput. Math. **15**, 23–37 (1984). https://doi.org/10.1080/00207168408803399

[112] Păun, G., Santean, L.: Parallel communicating grammar systems: the regular case. Ann. Univ. Buc., Ser. Matem.-Inform. **38**, 55–63 (1989)

[113] Rabin, M.O., Scott, D.: Finite automata and their decision problems. IBM J. Res. Dev. **3**, 114–125 (1959). https://doi.org/10.1147/rd.32.0114

[114] Sakoda, W.J., Sipser, M.: Nondeterminism and the size of two way finite automata. In: Proceedings of the Tenth Annual ACM Symposium on Theory of Computing (STOC 1978). pp. 275–286. ACM, ACM Press (1978). https://doi.org/10.1145/800133.804357

[115] Stearns, R.E.: A regularity test for pushdown machines. Inform. Control **11**, 323–340 (1967). https://doi.org/10.1016/S0019-9958(67)90591-8

[116] Valiant, L.G.: Regularity and related problems for deterministic pushdown automata. J. ACM **22**, 1–10 (1975). https://doi.org/10.1145/321864.321865

[117] Verlan, S.: On minimal context-free insertion-deletion systems. J. Autom. Lang. Comb. **12**, 317–328 (2007). https://doi.org/10.25596/jalc-2007-317

[118] Villagra, M., Yamakami, T.: Quantum state complexity of formal languages. In: Descriptional Complexity of Formal Systems (DCFS 2015). LNCS, vol. 9118, pp. 280–291. Springer (2015). https://doi.org/10.1007/978-3-319-19225-3_24

[119] Yamakami, T.: How does adiabatic quantum computation fit into quantum automata theory? Inf. Comput. **284**, 104694 (2022). https://doi.org/10.1016/j.ic.2021.104694

[120] Yu, S.: State complexity of regular languages. In: Dassow, J., Wotschke, D. (eds.) International Workshop on Descriptional Complexity of Automata, Grammars and Related Structures, Magdeburg, Germany, July 20 - 23, 1999. Preproceedings. vol. Preprint Nr. 17, pp. 77–88. Fakultät für Informatik, Universität Magdeburg, Magdeburg, Germany (1999)

[121] Yu, S., Zhuang, Q., Salomaa, K.: The state complexities of some basic operations on regular languages. Theor. Comput. Sci. **125**, 315–328 (1994). https://doi.org/10.1016/0304-3975(92)00011-F