

THE FORMAL LANGUAGE THEORY COLUMN

BY

GIOVANNI PIGHIZZINI

Dipartimento di Informatica
Università degli Studi di Milano
20135 Milano, Italy
`pighizzini@di.unimi.it`

Dear Reader, welcome to the Formal Language Theory Column!

Starting from this issue, I am the new editor of this column. First of all, I warmly thank my predecessor, Arto Salomaa, which was responsible of the column for many years. I would like to continue his work, by collecting contributions from the area that can be useful for exchanging new ideas, increasing and sharing the global knowledge in the field, and stimulating new researches. For instance, papers describing new developments, recent results, revisitations of classical topics as well as open problems will be welcome.

This column presents a survey paper written by Martin Kutrib and myself, on some recent trends in descriptonal complexity of formal languages.

RECENT TRENDS IN DESCRIPTIONAL COMPLEXITY OF FORMAL LANGUAGES

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

Giovanni Pighizzini
Dipartimento di Informatica
Università degli Studi di Milano
via Comelico 39, 20135 Milano, Italy
pighizzini@di.unimi.it

Abstract

Formal languages can be described by several means. A basic question is how succinctly can a descriptive system represent a formal language in comparison with other descriptive systems? What is the maximal size trade-off when changing from one system to another, and can it be achieved? Here, we select some recent trends in the descriptive complexity of formal languages and discuss the problems, results, and open questions. In particular, we present the main historical development and address the basic concepts of descriptive complexity from a general abstract perspective. Then we consider the representation by two-way finite automata, multi-head finite automata, and limited automata in more detail. Finally, we discuss a few further topics in note form. 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 formal languages by automata, grammars, equation systems, and other descriptive systems have been a subject of intensive research. The approach to analyze the size of systems as opposed to the computational power seems to originate from Stearns [64] who studied the relative succinctness of regular languages represented by deterministic finite automata 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 $t^{n^{n^t}}$. Later this triple exponential upper bound has been improved by one level of exponentiation in [66]. In the levels of exponentiation it is tight. In [50] a double exponential lower bound has been obtained. The precise bound is still an open problem. Probably the best-known result on descriptive complexity is the construction of a DFA that simulates a given nondeterministic finite automaton [60]. 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 [42, 50, 53]. Let us turn to another cornerstone of descriptive complexity theory in the seminal paper by Meyer and Fischer [50]. 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 descriptive system compared with the representation by an equivalent descriptor of the other descriptive system? In [50] 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*.

Nowadays, descriptive complexity has become a large and widespread area. Classical main branches not addressed in this summary are automata simulations, state complexity of operations, whose systematic study was initiated in [72], magic numbers, a research field initiated in [26], determinization of nondeterministic finite automata accepting subregular languages [3], transition complexity of NFA [6, 15, 22, 23, 41], and non-recursive trade-offs. Further results and references on these topics can be found, for example, in the surveys [13, 19, 20, 32].

1.1 Basic Concepts of Descriptive Complexity

In order to be more precise, we now turn to present and discuss the very basics of descriptive complexity.

We denote the set of nonnegative integers by \mathbb{N} . Let Σ^* denote the set of all words over a finite alphabet Σ . The *empty word* is denoted by λ , and we set $\Sigma^+ = \Sigma^* - \{\lambda\}$. For the *reversal of a word* w we write w^R and for its *length* 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 $L(X)$.

In order to be general, we first formalize the intuitive notion of a representation or description of a family of languages. A *descriptive 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 *descriptive system* \mathcal{S} is a set of finite descriptors such that each $D \in \mathcal{S}$ describes a formal language $L(D)$, and the underlying alphabet $\text{alph}(D)$ over which D represents a language can be obtained from D . The *family of languages represented* (or *described*) by \mathcal{S} is $L(\mathcal{S}) = \{L(D) \mid D \in \mathcal{S}\}$. For every language L , the set $\mathcal{S}(L) = \{D \in \mathcal{S} \mid L(D) = L\}$ is the set of its descriptors in \mathcal{S} . A *complexity measure* for a descriptive system \mathcal{S} is a total recursive mapping $c : \mathcal{S} \rightarrow \mathbb{N}$. From the viewpoint that a descriptive 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 such that their input alphabets can be extracted from the encodings. The set of these encodings is a descriptive system \mathcal{S} , and $L(\mathcal{S})$ is the family of regular languages.

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

In fact, we will use *length* to obtain a rough classification of different complexity measures. We distinguish between measures that (with respect to the size of the underlying alphabet) are recursively related with *length* and measures that are not. In the following, we only use complexity measures of the former type: If there is a total recursive function $g : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ such that, for all $D \in \mathcal{S}$, $\text{length}(D) \leq g(c(D), |\text{alph}(D)|)$, then c is said to be an *s-measure* (a *size measure*). Since for any coding alphabet there are only finitely many descriptors having at most length $g(c(D), |\text{alph}(D)|)$, over the same alphabet there are only finitely many descriptors in \mathcal{S} having the same size as D . If, in addition, for any alphabet Σ , the set of descriptors in \mathcal{S} describing languages over Σ is recursively enumerable in order of increasing size, then c is said to be an *sn-measure*. Clearly, *length*, *state*, and *trans* are sn-measures for finite automata.

Whenever we consider the relative succinctness of two descriptive systems \mathcal{S}_1 and \mathcal{S}_2 , we assume that the intersection $L(\mathcal{S}_1) \cap L(\mathcal{S}_2)$ is non-empty. Let \mathcal{S}_1 and \mathcal{S}_2 be descriptive systems with complexity measures c_1 and c_2 , respectively.

A total function $f : \mathbb{N} \rightarrow \mathbb{N}$, 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 $L(D_1) \in L(\mathcal{S}_2)$ 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 $f : \mathbb{N} \rightarrow \mathbb{N}$ 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 $L(D_1) \in L(\mathcal{S}_2)$, 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 sn-measures. That is, whenever the trade-off from one descriptive 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. (See, for example, [13, 16, 19, 32] for more on non-recursive trade-offs.)

2 Recent Trends

Over the years a lot of investigations of descriptonal complexity have been done documenting the importance of that field, its valuable concepts, and its vivid development. In the present section, we discuss only a few of the recent topics reflecting our personal view of what constitute the currently most interesting and challenging problems of descriptonal complexity theory.

2.1 Two-Way Finite Automata

Since the beginning of automata theory it is known that the possibility of moving the head on the input tape in both directions does not increase the computational power of finite automata, even if nondeterministic transitions are allowed [60, 62]. However, the devices so obtained, which are called *two-way finite automata*, can be smaller than equivalent one-way automata.

As a simple example, for each integer $n > 0$ let us consider the language $L_n = (a + b)^* a (a + b)^{n-1}$. We can easily build a two-way deterministic finite automaton (2DFA) which accepts it by scanning the entire input from left to right and, when the right end is reached, by moving to the left in order to verify whether or not the symbol in position n from the right is an a . This gives a 2DFA with $n + 2$ states, in contrast with the minimal DFA which requires 2^n states. Notice that L_n is accepted by a NFA with $n + 1$ states.

In 1978, Sakoda and Sipser [61] raised the question of the exact costs in states for the simulations of NFAs and 2NFAs by 2DFAs. They conjectured that these costs are not polynomial. To support such a conjecture, Sakoda and Sipser presented a complete analogy with the P versus NP question, by introducing a notion of reducibility between families of regular languages which allows to identify families of languages which are complete for these simulations. (For a detailed discussion on this approach we address the reader to the recent paper [27].) In spite of many attempts to solve it, the problem is still open.

However, in the last decade many progresses have been done by attacking and solving restricted versions of this question. In particular, in the literature three families of restrictions have been considered:

- restrictions on the simulating machines,

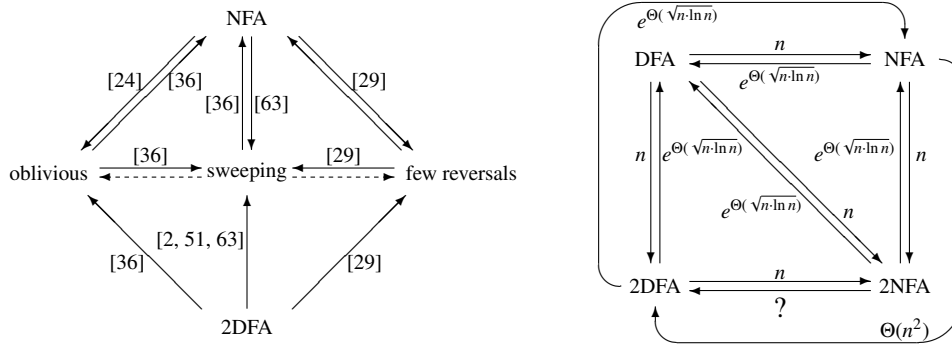


Figure 1: (*Left*) Costs of the simulation between different variants of automata. An arrow from a class A of machines to a class B indicates an exponential separation. A dashed arrow indicates a polynomial simulation. The (trivial) dashed arrow from oblivious, sweeping, and few reversal automata to 2DFAs are not depicted.

(*Right*) Costs of the *optimal* simulations between different kinds of *unary* automata [4, 49]. An arrow labeled $f(n)$ from a vertex A to a vertex B means that a unary n -state automaton in the class A can be simulated by an $f(n)$ -state automaton in the class B .

- restrictions on the class of languages,
- restrictions on the simulated machines.

We now briefly discuss some of these restrictions, referring the reader to [56] for a recent extended survey along these lines.

Exponential separations have been obtained for the simulation of NFAs and 2NFAs by the following restricted classes of 2DFAs:

- *Sweeping automata*: these devices can reverse the head direction only while visiting the endmarkers (two special symbols marking the left and the right ends of the input).
- *Oblivious automata*: the “trajectories” of the head on each two inputs of the same length should coincide, namely, the position of the input head at each step t of the computation does not depend on the input content, but *only on its length*.
- *Few reversal automata*: the number of reversals of the head direction is sublinear with respect to the input length.

However, all these separations cannot solve the general problem. In fact, it has also been proved that all these devices can require exponentially many states with respect to equivalent (unrestricted) 2DFAs. Those separations, with references to the literature, are summarized in Figure 1 (*Left*).

Concerning the second family of restrictions, interesting results have been found in the case of *unary languages*, namely languages defined over a one letter input alphabet. The state costs of the optimal simulations between different variants of unary automata

have been obtained by Chrobak [4] and by Mereghetti and Pighizzini [49], and they are summarized in Figure 1 (*Right*). From the picture we can observe that the cost of the optimal simulations in the unary case can be smaller than in the general case. For example, the cost of the simulation of n -state NFAs by DFAs reduces from 2^n to $e^{\Theta(\sqrt{n \cdot \ln n})}$. Quite surprisingly, eliminating at the same time both nondeterminism and two-way motion costs as eliminating only one of them. The question NFAs versus 2DFAs has been solved in the unary case in [4] by showing that the tight costs are polynomial, more precisely $\Theta(n^2)$. This gives also the best known lower bound for the general case.

Despite the unary case looks simpler than the general one, the question of 2NFAs versus 2DFAs not only is still open even in this case, but it seems also to be difficult and, at the same time, very challenging. First of all, we mention that in [11] it has been proved that each unary n -state 2NFA can be simulated by a 2DFA with $e^{O(\ln^2 n)}$ states. This gives a subexponential but still superpolynomial upper bound. Proving the optimality of this upper bound, or proving a smaller but still superpolynomial lower bound for the state cost of the simulation of *unary* 2NFAs by 2DFAs would separate deterministic and nondeterministic logarithmic space (L and NL, respectively). In fact, as showed in [12], if $L = NL$ then the state cost of the simulation of unary 2NFAs by 2DFAs is polynomial. The converse implication is also true if the classes are defined in a nonuniform way. For recent developments and further results we address the reader to [30].

Some extensions of the analysis for the unary case have been obtained by considering *outer nondeterministic automata*. These devices are 2NFAs that are restricted to take nondeterministic decisions *only* when the input head is scanning the endmarkers. Hence, transitions on “real” input symbols are deterministic. Notice that there are no restrictions on the head movements as for instance in sweeping automata. These models share several properties with unary 2NFAs [9, 30]. Among them, a subexponential state upper bound for the simulation by 2DFAs have been obtained and relationships with the question L vs. NL have been stated.

2.2 Multi-Head Finite Automata

Before we turn to present the known results, current studies, and open questions of descriptonal complexity issues of multi-head finite automata, we informally recall briefly what they are.

Let $k \geq 1$ be a natural number. A *nondeterministic two-way k -head finite automaton* (2NFA(k)) is a nondeterministic finite automaton having a single read-only input tape whose inscription is the input word in between two endmarkers. The k heads of the automaton can move freely on the tape but not beyond the endmarkers. A 2NFA(k) starts with all of its heads on the left endmarker. It halts when the transition function is not defined for the current situation. The input is accepted if and only if the automaton halts in an accepting state.

If in any case the transition function is either undefined or a singleton, then the k -head finite automaton is said to be *deterministic* ($2\text{DFA}(k)$). In case the heads never move to the left, the k -head finite automaton is said to be *one-way*. Nondeterministic and deterministic one-way k -head finite automata are denoted by $1\text{NFA}(k)$ and $1\text{DFA}(k)$.

Obviously, for one-head machines, regardless of whether they work one- or two-way, or of whether they are deterministic or nondeterministic, we obtain a characterization of the regular languages. On the other hand, a simple example is the non-context-free language $\{w_cw \mid w \in \{a, b\}^+\}$ that can be accepted by a deterministic one-way *two-head* finite automaton.

The power of multi-head finite automata is well studied in the literature (see, for example, [21] for a survey). The interest is also driven by the strong relation to the computational complexity classes L and NL . In fact, in [17] the characterizations $L = \bigcup_{k \geq 1} L(2\text{DFA}(k))$ and $NL = \bigcup_{k \geq 1} L(2\text{NFA}(k))$ are shown.

Taking a closer look reveals the natural questions for the descriptive and computational power of the *precise number of heads*. The questions for the computational power have eventually been answered in [52], where it is shown that, for each $k \geq 1$, there is a *unary* language accepted by some deterministic (nondeterministic) two-way finite automaton with $k + 1$ heads which is not accepted by any k -head deterministic (nondeterministic) two-way finite automaton, and in [71], where it is shown that the language

$$L_n = \{w_1\$w_2\$ \cdots \$w_{2n} \mid w_i \in \{a, b\}^* \text{ and } w_i = w_{2n+1-i}, \text{ for } 1 \leq i \leq n\}$$

can be accepted by a $1\text{DFA}(k)$ if and only if $n \leq \binom{k}{2}$. Thus, L_n can be used to separate the computational power of automata with $k + 1$ heads from those with k heads also in the one-way setting.

But how about the descriptive power? The question of determining the trade-offs between the levels of the head hierarchies arises immediately. It was Kapoutsis [28] who solved the problem for two-way machines. In particular, there are non-recursive trade-offs between all levels of the head hierarchies for deterministic and nondeterministic devices (cf. also [31]). Moreover, the enormous descriptive power of heads evolves already for *unary* languages.

Similarly, for one-way multi-head finite automata it is known [32] that the trade-offs between $1\text{DFA}(k + 1)$ and $1\text{DFA}(k)$, between $1\text{NFA}(k + 1)$ and $1\text{NFA}(k)$, and between $1\text{DFA}(k + 1)$ and $1\text{NFA}(k)$ are all non-recursive. Moreover, non-recursive trade-offs are shown between nondeterministic 2-head and deterministic k -head automata.

So, is the descriptive complexity of multi-head finite automata a fully developed area without major open problems? The answer is a little hidden. While the non-recursive trade-offs for two-way machines are already for unary languages, in the one-way case every accepted unary language boils down to a regular one. In [25, 65] it is shown that every unary language accepted by a one-way multi-head finite automaton

is semilinear and, thus, regular. So, a lot of new questions are arising from the fog. Just to say it with one sentence, all simulations between descriptive systems for unary regular languages and multi-head finite automata, all problems in connection with the descriptive complexity of language operations, as well as the size costs for simulating $k + 1$ -head by k -head automata are worth studying. First steps have been done in [38], where the following results are from. Here the complexity is measured by the number of states, that is, we use the measure *state*.

First, we turn to the number of states for the simulation of an n -state k -head finite automaton by a classical (one-head) deterministic or nondeterministic finite automaton. So, we consider the maximal head reduction. As is often the case in connection with unary languages, the function

$$F(n) = \max\{(c_1, c_2, \dots, c_l) \mid c_1, c_2, \dots, c_l \geq 1 \text{ and } c_1 + c_2 + \dots + c_l = n\},$$

plays a crucial role, where lcm denotes the *least common multiple*. It is well known that the c_i always can be chosen to be relatively prime. The function has been investigated by Landau [39, 40] who proved the asymptotic growth rate $\lim_{n \rightarrow \infty} \frac{\ln(F(n))}{\sqrt{n \cdot \ln n}} = 1$. The bounds $F(n) \in \Omega(e^{\sqrt{n \cdot \ln n}})$ and $F(n) \in O(e^{\sqrt{n \cdot \ln n}(1+o(1))})$ have been derived in [7].

For the simulation by a DFA, an upper bound of $O(n \cdot F(t \cdot n)^{k-1})$ and a lower bound of $n \cdot F(n)^{k-1}$ states is shown, where t is a constant depending only on k . Since both bounds are of order $e^{\Theta(\sqrt{n \cdot \ln n})}$, the trade-off for the simulation is tight in the order of magnitude. It is worth mentioning that for both bounds the number k of heads is a constant. It has been given as part of the bounds to be more precise.

For any constants $k \geq 2$ and prime $n \geq 2$, the lower bound is shown by construction of a unary n -state 1DFA(k) so that every equivalent deterministic or nondeterministic finite automaton has a cycle of at least $\{nc_i^{k-1} \mid 1 \leq i \leq l\} = n(c_1 c_2 \dots c_l)^{k-1} = n \cdot F(n)^{k-1}$ states.

Based on investigations of the length of words in unary languages accepted by n -state 1DFA(k), in [37] the upper bound is derived.

These results reveal that the costs for the simulation of 1DFA(k) by DFA are the same (in the order of magnitude) as for the simulation of NFA by DFA. From this point of view the two resources *heads* and *nondeterminism* are equally powerful. So the question for the costs of the mutual simulation of 1DFA(k) and NFA raises immediately. Trading k heads for nondeterminism is known to yield polynomially larger state sets, where the degree of the polynomial depends on k . For constants $k, n \geq 2$ any unary n -state 1DFA(k) can be simulated by some NFA with $O(n^{2k})$ states.

For the lower bound, the singleton languages $L_{k,n} = \{a^{(k-1)n^k}\}$, for $k, n \geq 2$, are used as witnesses, which are accepted by some n -state 1DFA(k). Clearly, any NFA accepting $L_{k,n}$ needs at least $(k-1)n^k + 1 \in \Omega(n^k)$ states to check that there is no shorter word accepted.

So far, we considered the costs for the head reduction. Next we turn to the converse question whether we can trade nondeterminism for heads, that is, we are interested in the state complexity for the NFA by 1DFA(k) simulation. Naturally, our upper

bound depends highly on the number k of heads available. If k is at least the (on first sight) cryptic number $t = \lfloor \frac{-3 + \sqrt{8n+1}}{2} \rfloor$, then the upper bound is quadratic, otherwise superpolynomial. Let $k \geq 1$, $n \geq 2$ be constants, $t = \lfloor \frac{-3 + \sqrt{8n+1}}{2} \rfloor$, and M be a unary n -state NFA. Then

$$n' \leq \begin{cases} n^2 - 2 + F(n), & \text{if } k = 1; \\ n^2 - 2 + \left(n - \frac{t^2+t}{2}\right)^{\lceil \frac{t}{k} \rceil}, & \text{if } 1 < k < t/2; \\ 2n^2, & \text{if } k \geq t/2. \end{cases}$$

states are sufficient for any equivalent 1DFA(k).

The lower bound reads as follows. Let $k \geq 1$ be a constant. For any integer $m \geq 1$

there is an integer $n > m$ and a unary n -state NFA, such that $c_2 \cdot \sqrt[k]{e^{\frac{\sqrt{2n}}{\sqrt{c_1 \ln(\sqrt{2n})}}}}$ states are necessary for any equivalent 1DFA(k), where $c_1, c_2 > 0$ are two constants.

So, there is a gap between upper and lower bound. It is currently a challenging open problem how to close this gap. Also open are the descriptive costs for simulations of unary *nondeterministic* one-way multi-head finite automata. Furthermore, how about the trade-offs between devices with $k + 1$ and k heads? Are these trade-offs gradually evolve to the maximal head reduction? Are there jumps from a certain level, say, from two heads to one head?

2.3 Limited Automata

Each class of the *Chomsky hierarchy* is defined in terms of grammars using some special kind of productions, where the form of the productions which are allowed for grammars of type $1 \leq k \leq 3$ is a restriction of the form used for grammars of type $k - 1$. From the point of view of language acceptors, each class of the hierarchy is characterized by a family of devices. However, while *linear bounded automata* used to characterize type 1 languages and *finite state automata* used to characterize type 3 languages can be seen as restrictions of (one-tape) Turing machines, which characterize type 0 languages, for type 2 languages, namely context-free, the characterization in terms of *pushdown automata* is usually presented. These devices are very useful to investigate and manipulate context-free languages. They also emphasize the main difference between regular and context-free languages, namely the possibility of representing recursive structures which, in terms of accepting devices, corresponds to increase the power of finite automata by adding a pushdown store. However, in a hierarchical view, pushdown automata do not appear as a special case of linear bounded automata.

Almost half a century ago, Hibbard discovered a different characterization of context-free languages, which uses a restricted version of Turing machines, called *scan limited automata* or, shortly, *limited automata* [18]. For each integer $d \geq 0$, a d -limited automaton (d -LA) is a two-way nondeterministic Turing machine which can rewrite the content of each tape cell *only in the first d visits*. He proved that, for each

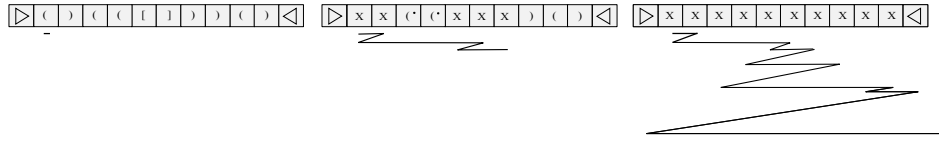


Figure 2: Some steps of the automaton A accepting the Dyck language D_k on input $()(())()$.

$d \geq 2$, the class of languages accepted by d -limited automata coincides with the class of context-free languages. Without affecting the computational power, these devices can be allowed to use only the part of the tape containing the input string. Hence, they are restrictions of linear bounded automata while, clearly, they are extensions of finite state automata. This gives a hierarchy of classes of Turing machines corresponding to the classes of the Chomsky hierarchy. Recently, in [57] this hierarchical view has been strengthened by proving that *deterministic* 2-limited automata characterize deterministic context-free language, solving in this way a problem which was left open by Hibbard.

Let us present, as a simple example, how the *Dyck language* D_k over the alphabet $\{(1,)_1, (2,)_2, \dots, (k,)_k\}$ of $k \geq 1$ types of brackets, namely the set of strings representing well balanced sequences of brackets, can be accepted by a deterministic 2-LA A . The automaton A starts scanning the tape until it finds a closing bracket $)_i$. Then, A substitutes $)_i$ with a symbol X and changes the head direction, moving to the left until it reaches an opening bracket $(_j$. If $i \neq j$ then A rejects. Otherwise, it writes X on the cell and changes again the head direction moving to the right, to search another closing bracket. This procedure is repeated as long as A does not reach one of the endmarkers (see Figure 2.) However, if the left endmarker is reached, then at least one of the closing brackets in the input w does not have a matching opening bracket. Hence, A rejects. On the other hand, if the right endmarker is reached, then A has to make sure that no unmatched opening brackets are left. In order to do this, it scans the entire tape from the right to the left and, if it finds an opening bracket which has not been rewritten, then it rejects. Otherwise, A accepts the input.

In [57] the equivalence between 2-LAs and PDAs has been revisited considering descriptonal complexity aspects. In particular, the following results have been obtained:

- Each 2-LA can be transformed into an equivalent PDA with an exponential increasing in the size. This gap cannot be reduced. Furthermore, the transformation preserves determinism.
- Conversely, each PDA can be transformed into an equivalent 2-LA of *polynomial* size. Even in this case, it is possible to preserve determinism.

Concerning determinism in d -limited automata for $d > 2$, it is not hard to see that the language $L = \{a^n b^n c \mid n \geq 0\} \cup \{a^n b^{2^n} d \mid n \geq 0\}$ is accepted by a deterministic 3-LA, which first completely traverses the input from left to right and then, depending

on the last input letter, starts to rewrite one or two bs for each a . However, L is not a deterministic context-free language. Hence, each 2-LA accepting it must be nondeterministic. As a matter of fact, Hibbard proved the existence of an infinite hierarchy of deterministic languages: for each integer $d > 2$ there is a language accepted by a deterministic d -LA which cannot be accepted by any deterministic $(d - 1)$ -LA. This result is also true for $d = 2$ but, in this case, it is a consequence of the fact that 1-limited automata accept only regular languages [67]. Descriptive complexity aspects of such equivalence have been recently investigated in [58].

Another characterization of context-free languages of a similar flavor has been obtained by Wechsung [68, 69]: for a fixed constant d , the machine is allowed to rewrite each input cell *only in the last d visits*.¹ Even in this case, for each $d \geq 2$ context-free languages are characterized, while for $d = 1$ the model is trivially equivalent to finite automata. The set of palindromes can easily be accepted in this model in a deterministic way with $d = 2$, just implementing the trivial algorithm which compares pair by pair the symbols in the corresponding positions starting from the left and right ends of the input and moving toward the middle. Any d -limited automaton accepting this language must perform nondeterministic choices, for each $d \geq 2$ [18]. On the other hand, the language $\{a^n b^{n+m} a^m \mid n, m \geq 0\}$ cannot be accepted in a deterministic way by the model proposed by Wechsung, for any $d \geq 2$ [54].

Finally, it is worth mentioning shortly what happens if we replace the constant d by a function $d(n)$ of the input length. In this case, both the models proposed by Hibbard and by Wechsung are equivalent to one-way auxiliary pushdown automata working in space $d(n)$, namely standard pushdown automata extended with a $d(n)$ space bounded work tape [70].

Probably the reader has noticed that even if this paper is devoted to recent trends in descriptive complexity of formal languages, in the above discussion on limited automata descriptive complexity aspects are very restricted. Actually, the investigation of descriptive complexity of limited automata has been an opportunity for revisiting and inspecting several aspects of these devices which are not so well known. We think that these devices deserve further investigation. Concerning their descriptive complexity, two points under investigations are the cost of the simulation of d -limited automata by pushdown automata for $d > 2$ and the cost of the simulations of d -limited automata in the unary case. It is worth remembering that, since unary context-free languages are regular, in the unary case the cost of the simulation of d -limited automata by finite automata should be also considered. Furthermore, both in the general and in the unary cases, descriptive complexity of d -limited automata with that of $(d - 1)$ -limited automata could be also compared, for each $d > 2$.

¹Wechsung introduced the term *return complexity* to indicate the maximum number of visits to a cell beginning with the first rewriting of its initial content. Hence, he considered machines with fixed return complexity d . In contrast, the maximum number of visits to a cell ending with the last rewriting of its content, namely the measure related to limited automata investigated by Hibbard, was called *dual return complexity*.

Both the notions proposed by Hibbard and Wechsung characterize, in different and in some sense dual ways, context-free language. Is it possible to obtain a more general notion of “limited automata” which captures these two notions, without increasing the computational power?

3 More Topics

We selected only a few topics in Section 2 which are close to our recent research interests. Of course, there are many other important “hot” topics that have recently been studied. We briefly mention some of them with references to the literature. However, our list is clearly far from being complete.

Unary Languages. The investigation of descriptonal complexity in the case of *unary* languages shows many different bounds with respect to the general case. Tools and properties from number theory have extensively been used. In the previous section we mentioned the unary case several times, see in particular Figure 1 (*Right*).

Restricted Pushdown Automata. Not so many results concerning descriptonal complexity of PDAs have been obtained. In the introduction we mentioned the non-recursive trade-off between context-free grammars, or equivalently PDAs, for regular languages and finite automata, and the double exponential gap between deterministic PDAs accepting regular languages and finite automata. Both conversions have been investigated in the unary case obtaining optimal recursive bounds [59, 55].

Two other connections between PDAs and regular languages have been recently considered with respect to descriptonal complexity: PDAs with pushdown stores of constant height [1, 8]; languages consisting of all pushdown contents in accepting computations of PDAs [46, 10] (these languages are known to be regular [14]).

Two-Way Pushdown Automata. While for finite state automata the possibility of moving the input head in both directions does not increase the computational power, it is well-known that this is not true in the case of PDAs. For example, the non-context-free language $\{a^n b^n c^n \mid n \geq 0\}$ can easily be accepted by a *two-way pushdown automaton*. Up to now, the computational power of these models has not been clearly identified. For example, it is unknown if they are equivalent to linear bounded automata and if the nondeterministic variant is more powerful than the deterministic one. Recent descriptonal complexity results concerning these models and taking into account, besides the size of the devices, the number of the head reversals and the number of turns of the pushdown store, have been presented in [48].

Cellular Automata and Iterative Arrays. These devices are often studied as massively parallel language acceptors [33]. The investigation of their descriptonal complexity originates in [43, 45]. It turned out that in many cases the resources given to cellular automata in connection with massiveness yield non-recursive trade-offs. In the recent papers [35, 47] it is shown that even very little additional resources

have a big impact on the necessary size of the devices. For example, adding sublinearly more time obtaining time complexities strictly in between real time and linear time, adding dimensions, allow the communication cell to perform a few nondeterministic steps, or increase the number of bits that may be communicated to neighboring cells slightly, allows arbitrary savings in the size of the descriptions of the arrays which cannot be bounded by any computable function. So the challenging tasks are to identify resources that can be added to or modifications of cellular automata that yield *recursive* trade-offs. Examples are the decompositions and generalized presentation of languages gained in language expressions with operations under which the family in question is closed, the so-called operational state complexity (see [34] for results on one-way cellular automata). Another approach which bounds the number of cells available can be found in [44].

References

- [1] Bednárová, Z., Geffert, V., Mereghetti, C., Palano, B.: The size-cost of Boolean operations on constant height deterministic pushdown automata. *Theoret. Comput. Sci.* 449, 23–36 (2012)
- [2] Berman, P.: A note on sweeping automata. In: *International Colloquium on Automata, Languages and Programming (ICALP 1980)*. LNCS, vol. 85, pp. 91–97. Springer (1980)
- [3] Bordihn, H., Holzer, M., Kutrib, M.: Determinization of finite automata accepting subregular languages. *Theoret. Comput. Sci.* 410, 3209–3222 (2009)
- [4] Chrobak, M.: Finite automata and unary languages. *Theoret. Comput. Sci.* 47(2), 149–158 (1986), errata: [5]
- [5] Chrobak, M.: Errata to “Finite automata and unary languages”. *Theoret. Comput. Sci.* 302, 497–498 (2003)
- [6] Domaratzki, M., Salomaa, K.: Lower bounds for the transition complexity of NFAs. In: *Mathematical Foundations of Computer Science (MFCS 2006)*. LNCS, vol. 4162, pp. 315–326. Springer (2006)
- [7] Ellul, K.: *Descriptive Complexity Measures of Regular Languages*. Master’s thesis, University of Waterloo, Canada (2004)
- [8] Geffert, V., Bednárová, Z., Mereghetti, C., Palano, B.: Boolean language operations on nondeterministic automata with a pushdown of constant height. In: *Computer Science Symposium in Russia (CSR 2013)*. LNCS, vol. 7913, pp. 100–111. Springer (2013)
- [9] Geffert, V., Guillon, B., Pighizzini, G.: Two-way automata making choices only at the endmarkers. In: *Language and Automata Theory and Applications (LATA 2012)*, LNCS, vol. 7183, pp. 264–276. Springer (2012)
- [10] Geffert, V., Malcher, A., Meckel, K., Mereghetti, C., Palano, B.: A direct construction of finite state automata for pushdown store languages. In: *Descriptive Complexity of Formal Systems (DCFS 2013)*. LNCS, vol. 8031, pp. 90–101. Springer (2013)

- [11] Geffert, V., Mereghetti, C., Pighizzini, G.: Converting two-way nondeterministic unary automata into simpler automata. *Theoret. Comput. Sci.* 295, 189–203 (2003)
- [12] Geffert, V., Pighizzini, G.: Two-way unary automata versus logarithmic space. *Inform. Comput.* 209, 1016–1025 (2011)
- [13] Goldstine, J., Kappes, M., Kintala, C.M.R., Leung, H., Malcher, A., Wotschke, D.: Descriptive complexity of machines with limited resources. *J. UCS* 8, 193–234 (2002)
- [14] Greibach, S.A.: A note on pushdown store automata and regular systems. *Proc. Amer. Math. Soc.* 18, 263–268 (1967)
- [15] Gruber, H., Holzer, M.: On the average state and transition complexity of finite languages. *Theoret. Comput. Sci.* 387, 155–166 (2007)
- [16] Gruber, H., Holzer, M., Kutrib, M.: On measuring non-recursive trade-offs. *J. Autom., Lang. Comb.* 15, 107–120 (2010)
- [17] Hartmanis, J.: On non-determinacy in simple computing devices. *Acta Inform.* 1, 336–344 (1972)
- [18] Hibbard, T.N.: A generalization of context-free determinism. *Inform. Control* 11, 196–238 (1967)
- [19] Holzer, M., Kutrib, M.: Descriptive complexity – An introductory survey. In: *Scientific Applications of Language Methods*, pp. 1–58. Imperial College Press (2010)
- [20] Holzer, M., Kutrib, M.: Descriptive and computational complexity of finite automata – A survey. *Inform. Comput.* 209, 456–470 (2011)
- [21] Holzer, M., Kutrib, M., Malcher, A.: Complexity of multi-head finite automata: Origins and directions. *Theoret. Comput. Sci.* 412, 83–96 (2011)
- [22] Hromkovič, J., Schnitger, G.: NFAs with and without ϵ -transitions. In: *International Colloquium on Automata, Languages and Programming (ICALP 2005)*. LNCS, vol. 3580, pp. 385–396. Springer (2005)
- [23] Hromkovič, J., Seibert, S., Wilke, T.: Translating regular expressions into small ϵ -free nondeterministic finite automata. *J. Comput. System Sci.* 62, 565–588 (2001)
- [24] Hromkovič, J., Schnitger, G.: Nondeterminism versus determinism for two-way finite automata: Generalizations of Sipser’s separation. In: *International Colloquium on Automata, Languages and Programming (ICALP 2003)*. LNCS, vol. 2719, pp. 439–451. Springer (2003)
- [25] Ibarra, O.H.: A note on semilinear sets and bounded-reversal multihead pushdown automata. *Inform. Process. Lett.* 3, 25–28 (1974)
- [26] Iwama, K., Kambayashi, Y., Takaki, K.: Tight bounds on the number of states of DFAs that are equivalent to n -state NFAs. *Theoret. Comput. Sci.* 237, 485–494 (2000)
- [27] Kapoutsis, C.: Minicomplexity. In: *Descriptive Complexity of Formal Systems (DCFS 2012)*, LNCS, vol. 7386, pp. 20–42. Springer (2012)
- [28] Kapoutsis, C.A.: Non-recursive trade-offs for two-way machines. *Int. J. Found. Comput. Sci.* 16, 943–956 (2005)

- [29] Kapoutsis, C.A.: Nondeterminism is essential in small two-way finite automata with few reversals. *Inform. Comput.* 222, 208–227 (2013)
- [30] Kapoutsis, C.A., Pighizzini, G.: Two-way automata characterizations of L/poly versus NL. In: *Computer Science Symposium in Russia (CSR 2012)*. LNCS, vol. 7353, pp. 217–228. Springer (2012)
- [31] Kutrib, M.: On the descriptive power of heads, counters, and pebbles. *Theoret. Comput. Sci.* 330, 311–324 (2005)
- [32] Kutrib, M.: The phenomenon of non-recursive trade-offs. *Int. J. Found. Comput. Sci.* 16, 957–973 (2005)
- [33] Kutrib, M.: Cellular automata and language theory. In: *Encyclopedia of Complexity and System Science*, pp. 800–823. Springer (2009)
- [34] Kutrib, M., Lefèvre, J., Malcher, A.: The size of one-way cellular automata. In: *Cellular Automata and Discrete Complex Systems (AUTOMATA 2010)*. pp. 71–90. *Discrete Mathematics and Theoretical Computer Science Proceedings, DMTCS* (2010)
- [35] Kutrib, M., Malcher, A.: The size impact of little iterative array resources. *J. Cellular Automata* 7, 489–507 (2012)
- [36] Kutrib, M., Malcher, A., Pighizzini, G.: Oblivious two-way finite automata: Decidability and complexity. In: *Latin 2012: Theoretical Informatics*, LNCS, vol. 7256, pp. 208–221. Springer (2012)
- [37] Kutrib, M., Malcher, A., Wendlandt, M.: States and heads do count for unary multi-head finite automata. In: *Developments in Language Theory (DLT 2012)*. LNCS, vol. 7410, pp. 214–225. Springer (2012)
- [38] Kutrib, M., Malcher, A., Wendlandt, M.: Size of unary one-way multi-head finite automata. In: *Descriptive Complexity of Formal Systems (DCFS 2013)*. LNCS, vol. 8031, pp. 148–159. Springer (2013)
- [39] Landau, E.: Über die Maximalordnung der Permutationen gegebenen Grades. *Archiv der Math. und Phys.* 3, 92–103 (1903)
- [40] Landau, E.: *Handbuch der Lehre von der Verteilung der Primzahlen*. Teubner, Leipzig (1909)
- [41] Lifshits, Y.: A lower bound on the size of ϵ -free NFA corresponding to a regular expression. *Inform. Process. Lett.* 85, 293–299 (2003)
- [42] 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, 328–335 (1966)
- [43] Malcher, A.: Descriptive complexity of cellular automata and decidability questions. *J. Autom., Lang. Comb.* 7, 549–560 (2002)
- [44] Malcher, A.: On one-way cellular automata with a fixed number of cells. *Fund. Inform.* 58, 355–368 (2003)
- [45] Malcher, A.: On the descriptive complexity of iterative arrays. *IEICE Trans. Inf. Syst.* E87-D(3), 721–725 (2004)

- [46] Malcher, A., Meckel, K., Mereghetti, C., Palano, B.: Descriptive complexity of pushdown store languages. In: *Descriptive Complexity of Formal Systems (DCFS 2012)*. LNCS, vol. 7386, pp. 209–221. Springer (2012)
- [47] Malcher, A., Mereghetti, C., Palano, B.: Sublinearly space bounded iterative arrays. *Int. J. Found. Comput. Sci.* 21, 843–858 (2010)
- [48] Malcher, A., Mereghetti, C., Palano, B.: Descriptive complexity of two-way pushdown automata with restricted head reversals. *Theoret. Comput. Sci.* 449, 119–133 (2012)
- [49] Mereghetti, C., Pighizzini, G.: Optimal simulations between unary automata. *SIAM J. Comput.* 30, 1976–1992 (2001)
- [50] 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)
- [51] Micali, S.: Two-way deterministic finite automata are exponentially more succinct than sweeping automata. *Inform. Process. Lett.* 12, 103–105 (1981)
- [52] Monien, B.: Two-way multihead automata over a one-letter alphabet. *RAIRO Inform. Théor.* 14, 67–82 (1980)
- [53] 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(10), 1211–1214 (1971)
- [54] Peckel, J.: On a deterministic subclass of context-free languages. In: *Mathematical Foundations of Computer Science (MFCS 1977)*. LNCS, vol. 53, pp. 430–434. Springer (1977)
- [55] Pighizzini, G.: Deterministic pushdown automata and unary languages. *Int. J. Found. Comput. Sci.* 20(4), 629–645 (2009)
- [56] Pighizzini, G.: Two-way finite automata: Old and recent results. *Fund. Inform.* 126(2-3), 225–246 (2013)
- [57] Pighizzini, G., Pisoni, A.: Limited automata and context-free languages. In: *Non-Classical Models of Automata and Applications (NCMA 2013)*. books@ocg.at, vol. 294, pp. 209–223. Austrian Computer Society, Vienna (2013)
- [58] Pighizzini, G., Pisoni, A.: Limited automata and regular languages. In: *Descriptive Complexity of Formal Systems (DCFS 2013)*. LNCS, vol. 8031, pp. 253–264. Springer (2013)
- [59] Pighizzini, G., Shallit, J., Wang, M.W.: Unary context-free grammars and pushdown automata, descriptive complexity and auxiliary space lower bounds. *J. Comput. System Sci.* 65, 393–414 (2002)
- [60] Rabin, M.O., Scott, D.: Finite automata and their decision problems. *IBM J. Res. Dev.* 3, 114–125 (1959)
- [61] Sakoda, W.J., Sipser, M.: Nondeterminism and the size of two way finite automata. In: *Symposium on Theory of Computing (STOC 1978)*, pp. 275–286. ACM Press, New York (1978)
- [62] Shepherdson, J.C.: The reduction of two-way automata to one-way automata. *IBM J. Res. Dev.* 3, 198–200 (1959)

- [63] Sipser, M.: Lower bounds on the size of sweeping automata. *J. Comput. System Sci.* 21, 195–202 (1980)
- [64] Stearns, R.E.: A regularity test for pushdown machines. *Inform. Control* 11, 323–340 (1967)
- [65] Sudborough, I.H.: Bounded-reversal multihead finite automata languages. *Inform. Control* 25, 317–328 (1974)
- [66] Valiant, L.G.: Regularity and related problems for deterministic pushdown automata. *J. ACM* 22, 1–10 (1975)
- [67] Wagner, K., Wechsung, G.: *Computational Complexity*. Reidel, Dordrecht (1986)
- [68] Wechsung, G.: Characterization of some classes of context-free languages in terms of complexity classes. In: *Mathematical Foundations of Computer Science (MFCS 1975)*. LNCS, vol. 32, pp. 457–461. Springer (1975)
- [69] Wechsung, G.: Kompliziertheitstheoretische Charakterisierung der kontextfreien und linearen Sprachen. *J. Inform. Process. Cybern.* 12(6), 289–300 (1976)
- [70] Wechsung, G., Brandstädt, A.: A relation between space, return and dual return complexities. *Theoret. Comput. Sci.* 9, 127–140 (1979)
- [71] Yao, A.C., Rivest, R.L.: $k + 1$ heads are better than k . *J. ACM* 25, 337–340 (1978)
- [72] Yu, S.: State complexity of regular languages. *J. Autom., Lang. Comb.* 6, 221–234 (2001)