# A Circuit Complexity Approach to Transductions

Michaël Cadilhac[1(✉)], Andreas Krebs[1],
Michael Ludwig[1], and Charles Paperman[2]

[1] Wilhelm Schickard Institut, Universität Tübingen, Tübingen, Germany
michael@cadilhac.name, mail@krebs-net.de,
ludwigm@informatik.uni-tuebingen.de
[2] University of Warsaw and Warsaw Center of Mathematics
and Computer Science, Warsaw, Poland
Charles.Paperman@liafa.univ-paris-diderot.fr

**Abstract.** Low circuit complexity classes and regular languages exhibit very tight interactions that shade light on their respective expressiveness. We propose to study these interactions at a functional level, by investigating the deterministic rational transductions computable by constant-depth, polysize circuits. To this end, a circuit framework of independent interest that allows variable output length is introduced. Relying on it, there is a general characterization of the set of transductions realizable by circuits. It is then decidable whether a transduction is definable in $\mathrm{AC}^0$ and, assuming a well-established conjecture, the same for $\mathrm{ACC}^0$.

## Introduction

The regular languages in circuit complexity classes play an instrumental role in some of the most emblematic results of circuit complexity. The celebrated result of Furst, Saxe and Sipser [11] shows that the regular language $\mathrm{PARITY} = \{w \in \{0,1\}^* \mid |w|_1 \equiv 0 \mod 2\}$ is not in $\mathrm{AC}^0$, the class of constant-depth, polysize, unbounded fan-in circuits. As PARITY belongs to $\mathrm{ACC}^0$ (which allows in addition unbounded fan-in modulo gates), this separates $\mathrm{AC}^0$ and $\mathrm{ACC}^0$. Barrington's theorem [1] states that the regular languages are complete for the class $\mathrm{NC}^1$ of logdepth, polysize, and constant fan-in circuits. Further, Koucký, Pudlák, and Thérien [12] show that regular languages separate classes defined by $\mathrm{ACC}^0$ circuits using linear number of gates and using linear number of wires.

The classification of regular languages within circuit complexity classes thus attracted interest, culminating in the results of Barrington *et al.* [2] that entirely describe the regular languages in $\mathrm{AC}^0, \mathrm{ACC}^0$ and $\mathrm{NC}^1$. The algebraic property of regular languages studied therein deviates sharply from the prevailing line of work at the time, which relied on the study of the syntactic monoids of regular languages. (The syntactic monoid is the monoid of transformations of states of the minimal automaton.) Indeed, $\mathrm{PARITY} \notin \mathrm{AC}^0$, while the language EVEN of even-length words over $\{0,1\}$, which has the same syntactic monoid, does


G.F. Italiano et al. (Eds.): MFCS 2015, Part I, LNCS 9234, pp. 141–153, 2015.
DOI: 10.1007/978-3-662-48057-1_11

belong to $\mathrm{AC}^0$. Hence the class of regular languages in $\mathrm{AC}^0$ does not admit a characterization solely in terms of the syntactic monoids.

We propose to take this study to the functional case, that is, to characterize the functions realized by rational transducers (i.e., input/output automata) that are expressible by an $\mathrm{AC}^0$ circuit family. Similarly to the context at the time of [2], we face a situation where, to the best of our knowledge, most characterizations focused on algebraic properties that would blur the line between PARITY and EVEN (e.g., [14]).

We rely on a property we call *continuity* for a class of languages $\mathcal{V}$, as borrowed from the field of topology: a transduction $\tau$ is $\mathcal{V}$-continuous if it preserves $\mathcal{V}$ by inverse image (i.e., $\forall L \in \mathcal{V}, \tau^{-1}(L) \in \mathcal{V}$). It is well known that any transduction $\tau$ is continuous for the regular languages; together with an additional property on the output length of $\tau$, this even characterizes deterministic transductions [3]. Namely, with $d(u, v) = |u| + |v| - |u \wedge v|$, where $u \wedge v$ is the largest common prefix of $u$ and $v$, the latter property is that $d(\tau(u), \tau(v)) \leq k \times d(u, v)$, a strong form of *uniform* continuity. Continuity thus appears as a natural invariant when characterizing transductions—the *forward* behaviors of $\tau$, that is, its images, are less relevant, as any NP problem is the image of $\Sigma^*$ under an $\mathrm{AC}^0$ function [4]. Our contributions are three-fold:

- We propose a model of circuits that allows for functions of unrestricted output length: as opposed to previous models, e.g., [19], we do not impose the existence of a mapping between the input and output lengths.
- Relying on this model, we characterize the deterministic rational transductions computed by $\mathrm{AC}^0$ circuits with access to gates in a given class. This characterization relies for one part on algebraic objects similar to the ones used in [2], through the use of the modern framework of *lm-varieties* [18]. For the other part, we rely on the notion of *continuity*. This bears a striking resemblance to the characterization of Reutenauer and Schützenberger [16] of the transductions with a group as transition monoid.
- The characterization then leads to the decidability of the membership of a deterministic rational transduction in $\mathrm{AC}^0$ or in $\mathrm{ACC}^0$. This is effective in the sense that an appropriate circuit can be produced realizing the transduction.

In Sect. 1, we succinctly cover the automata- and circuit-theoretic notions necessary to our presentation. In Sect. 2, we introduce the circuit model for variable output length functions and argue for its legitimacy. In Sect. 3, we show that studying the transition morphism of an automaton is equivalent to studying the languages accepted at each of its states; this enables us to keep to a minimum the algebraic references throughout our presentation. In Sect. 4, we show the aforementioned characterization and delay to Sect. 5 its implications on $\mathrm{AC}^0$ and $\mathrm{ACC}^0$. We discuss the results and their limitations in Sect. 6.

## 1 Preliminaries

*Monoid, morphisms, quotient.* A monoid is a set equipped with a binary associative operation, denoted multiplicatively, with a unit element. For an alphabet $\Sigma$,

the set $\Sigma^*$ is the free monoid generated by $\Sigma$, its unit element being the empty word $\varepsilon$. A morphism is a map $\varphi\colon M \to N$ satisfying $\varphi(ab) = \varphi(a)\varphi(b)$ and $\varphi(1) = 1$, with $a, b \in M$ and 1 denoting the unit element of $M$ and $N$. A morphism $\varphi\colon \Sigma^* \to T^*$ is an *lm-morphism*, where lm stands for *length-multiplying*, if there is a $k$ such that $\varphi(\Sigma) \subseteq T^k$. Given a language $L$ and a word $u$, the *left quotient* of $L$ by $u$ is the set $u^{-1}L = \{v \mid uv \in L\}$. The *right quotient* $Lu^{-1}$ is defined symmetrically. For $w \in \Sigma^*$ and $a \in \Sigma$, we let $|w|_a$ be the number of $a$'s in $w$, i.e., the image of $w$ under the morphism $a \mapsto 1$, $\Sigma \setminus \{a\} \mapsto 0$ into $(\mathbb{N}, +)$.

*Circuits.* We use standard notations, as presented for instance in [17] and [19]. By $\mathrm{AC}^0$, we denote the class of languages recognized by constant-depth, polysize circuit families with Boolean gates of unbounded fan-in. We consider *nonuniform* families, that is, we leave unconstrained the mapping from the input size $n$ to the circuit with $n$ inputs. Such families recognize languages in $L \subseteq \{0,1\}^*$; to extend this to any alphabet $\Sigma$, we always assume there is a canonical map from $\Sigma$ to $\{0,1\}^{|\Sigma|}$, that lets us encode and decode words of $\Sigma^*$ in binary. A language $L$ naturally defines an $L$-gate which outputs 1 iff its input is in $L$; for instance, $\{0,1\}^*1\{0,1\}^*$ defines the OR gate. For a class of languages $\mathcal{V}$, we write $\mathrm{AC}^0(\mathcal{V})$ for languages recognized by $\mathrm{AC}^0$ circuit families with access to $L$-gates for all $L \in \mathcal{V}$. We let $\mathrm{ACC}^0 = \mathrm{AC}^0(\mathrm{MOD})$ where MOD is the class of regular languages on $\{0,1\}^*$ of the form $\{|w|_1 \equiv 0 \mod k\}$ for some $k$. Further, we define $\mathrm{TC}^0 = \mathrm{AC}^0(\mathrm{MAJ})$ where MAJ is the nonregular language $\{|w|_1 \geq |w|_0 \mid w \in \{0,1\}^*\}$. We will occasionally rely on the conjectured and widely-believed separation of $\mathrm{ACC}^0$ and $\mathrm{TC}^0$. Extending circuits to functions, a function $f$ is in $\mathrm{FAC}^0$ if there is a family of constant-depth, polysize circuits with multiple ordered output bits, such that $f(u)$ is the output of the circuit for input size $|u|$. We naturally extend the notation $\mathrm{AC}^0(\mathcal{V})$ to $\mathrm{FAC}^0(\mathcal{V})$.

*Automata.* A *deterministic automaton* is a tuple $A = (Q, \Sigma, \delta, q_0, F)$, where $Q$ is the finite set of states, $\Sigma$ the alphabet, $\delta\colon Q \times \Sigma \to Q$ is a partial transition function, $q_0$ is the initial state, and $F$ is the set of final states. We naturally extend $\delta$ to words by letting $\delta(q, \varepsilon) = q$, and $\delta(q, aw) = \delta(\delta(q, a), w)$ when $\delta(q, a)$ is defined. We always assume that any state $q$ is accessible and coaccessible, i.e., there is a word $uv$ such that $\delta(q_0, u) = q$ and $\delta(q, v) \in F$. We write $L(A, q)$ for $\{w \mid \delta(q_0, w) = q\}$, and $L(A) = \cup_{f \in F} L(A, f)$ for the language of $A$. For two states $q, q'$, we say that $q$ can be *separated* from $q'$ in $\mathcal{V}$ if there is a language $L$ in $\mathcal{V}$ such that $L(A, q) \subseteq L \subseteq \overline{L(A, q')}$. An automaton is *all-separable in* $\mathcal{V}$ if each pair of distinct states can be separated in $\mathcal{V}$. It is *all-definable in* $\mathcal{V}$ if every language $L(A, q)$ is in $\mathcal{V}$. We often use the shorter terms $\mathcal{V}$-all-separable and $\mathcal{V}$-all-definable, of self-explanatory meanings. We write REG for the class of regular languages.

*Continuity, Lm-varieties.* A mapping $f\colon \Sigma^* \to T^*$ is *continuous for* $\mathcal{V}$, in short $\mathcal{V}$*-continuous*, if $L \in \mathcal{V}$ implies $f^{-1}(L) \in \mathcal{V}$—this name stems from the notion of continuity in topology. The sets of regular languages recognized by circuit families form a backbone of our work. It is thus natural to assume that these

sets be closed under operations that $\mathrm{AC}^0$ circuits can compute; this is formalized as follows. A class of languages $\mathcal{V}$ is an *lm-variety* if it is a Boolean algebra of languages closed under left and right quotient such that any lm-morphism is $\mathcal{V}$-continuous. It can be shown that if $\mathrm{AC}^0(\mathcal{V}) \cap \mathrm{REG} = \mathcal{V}$, then $\mathcal{V}$ is an lm-variety. As is customary, we write $\mathcal{QA}$ for $\mathrm{AC}^0 \cap \mathrm{REG}$ and $\mathcal{M}_{\mathrm{sol}}$ for $\mathrm{ACC}^0 \cap \mathrm{REG}$—these names stem from the algebraic classes recognizing the languages: quasi-aperiodic stamps and solvable monoids respectively, see Sect. 3 and [17] for more details. In particular, $\mathrm{ACC}^0 \neq \mathrm{TC}^0$ iff $\mathrm{AC}^0(\mathcal{M}_{\mathrm{sol}}) \cap \mathrm{REG} = \mathcal{M}_{\mathrm{sol}}$ [2]. In the sequel, the symbol $\mathcal{V}$ always denotes some lm-variety of languages.

*Transducers.* A *deterministic transducer* is a tuple $A = (Q, \Sigma, T, \delta, \nu, q_0, F)$ which is an automaton equipped with an additional alphabet $T$ and a mapping $\nu\colon Q \times \Sigma \to T^*$ of same domain as $\delta$. We extend $\nu$ to words in $\Sigma^*$ by letting $\nu(q, \varepsilon) = \varepsilon$ and $\nu(q, aw) = \nu(q, a)\nu(\delta(q, a), w)$, when $\delta(q, a)$ is defined. The partial function $\tau\colon \Sigma^* \to T^*$ mapping $w \in L(A)$ to $\nu(q_0, w)$ is called a *transduction*. A transducer is said to be *output-minimal* if for every pair of states $q, q'$, there is a word $w$ such that either only one of $\delta(q, w)$ or $\delta(q', w)$ is final, or both are and $\nu(q, w) \neq \nu(q', w)$. For any transduction $\tau$, we fix an arbitrary output-minimal transducer $\mathsf{MinT}(\tau)$ realizing it. Note that given a transducer, one can easily compute an output-minimal transducer realizing the same transduction. We will see that the choice of $\mathsf{MinT}(\tau)$ does not bear any impact on the results.

*We freely use $Q, \Sigma, T$, etc. when an automaton or a transducer is under study, with the understanding that they are the relevant components of its defining tuple. Our focus being solely on automata, transducers, and transductions that are deterministic, we will omit mentioning determinism from now on.*

## 2 Circuit Frameworks for Variable-Length Functions

In the literature, most of the work on functions computed by circuits focus on variants of the class $\mathrm{FAC}^0$ (see, e.g., [19]). In these, multiple (ordered) output gates are provided, and there is thus an implicit mapping from input length to output length. Towards circumventing this limitation, we propose a few different frameworks, and establish some formal shortcomings in order to legitimize our final choice. Our main requirement is that functions defined using constant-depth, polysize circuits should be $\mathrm{AC}^0$-continuous—this corresponds to a simple composition of the circuits. In particular, $\mathrm{FAC}^0$ functions are $\mathrm{AC}^0$-continuous.

### 2.1 Noninversability

We first consider circuits with a pair of inputs $\langle u, v\rangle$, where the represented function is valued $v$ on $u$ if the circuit accepts the pair $\langle u, v\rangle$. By making no syntactic distinction between input and output, any function has the same complexity as its inverse if it is functional. We show that this blurs definability:

**Proposition 1.** *There is an $\mathrm{AC}^0$-continuous transduction in $\mathrm{FAC}^0$ whose inverse is functional and not $\mathrm{AC}^0$-continuous.*

*Proof.* Consider the minimal, two-state automaton for $L = 0^*(a0^*b0^*)^*$ and turn it into a transducer by letting $\nu(\cdot, 0) = 0$ and $\nu(\cdot, a) = \nu(\cdot, b) = 1$, and call $\tau$ the resulting transduction. The $\mathrm{FAC}^0$ circuit for $\tau$ first checks that the input is in $L$. This can be done as $L \in \mathrm{AC}^0$, a fact that can be seen relying on the logical characterization of $\mathrm{AC}^0$: a word is in $L$ iff its first non-0 letter is an $a$, its last a $b$, and the closest non-0 letters to an $a$ (resp. a $b$) are $b$'s (resp. $a$'s). Next, the circuit simply maps 0 to 0 and $a, b$ to 1. The transduction being in $\mathrm{FAC}^0$, it is $\mathrm{AC}^0$-continuous. Now let $\sigma = \tau^{-1}$, it is clearly functional. But $\sigma^{-1}(L)$ is PARITY, hence $\sigma$ is not $\mathrm{AC}^0$-continuous. □

Thus, much in the fashion of $\mathrm{FAC}^0$, this implies that there should be distinguished input and output gates. We next deal with how their lengths are specified.

### 2.2 Output Length as a Parameter

Aiming for a natural and succinct model, we may want that the family of circuits be parametrized solely by the input length. In such a framework, the presented circuit for a given input length is equipped with a way to "deactivate" output gates, in order to allow for different output lengths. Formalizing this idea further, a *deactivating circuit* $C$ with $n$ inputs and $m$ outputs is an usual circuit with an extra input valued $\mathbf{z}$, a new constant symbol. This new symbol behaves as follows: $1 \vee \mathbf{z} = \mathbf{z} \vee 1 = 1$, and any other combination of $\mathbf{z}$ with $0, 1, \vee, \wedge, \neg$ is valued $\mathbf{z}$. The output of $C$ on a given input is its usual output stripped of the $\mathbf{z}$ symbol. The frameworks used in [5,10,14] are logic counterparts of this model. Then:

**Proposition 2.** *There is a transduction expressible as a constant-depth, polysize family of deactivating circuits which is not* $\mathrm{AC}^0$*-continuous.*

*Proof.* The erasing morphism $0 \mapsto \varepsilon$, $1 \mapsto 1$ is a transduction $\tau$ that can be expressed as a family of circuits as in the statement of the Proposition, but $\tau^{-1}(1^{2\mathbb{N}})$ is PARITY $\notin \mathrm{AC}^0$. □

We thus reach the following definition, that will serve as a basis for our study:

**Definition 1 (Functional Circuits).** *A function* $\tau\colon \Sigma^* \to T^*$ *is expressed as a circuit family* $(C_m^n)_{n,m\geq 0}$*, where* $C_m^n$ *is a circuit with* $n$ *inputs and* $m+1$ *outputs, if:*

$$(\forall u, v \in \Sigma^*) \qquad \tau(u) = v \Leftrightarrow C_{|v|}^{|u|}(u) = (v, 1) \ .$$

*The* size *of the family is the mapping from* $\mathbb{N}$ *to* $\mathbb{N} \cup \{\infty\}$*, defined by* $n \mapsto \sup_{m\geq 0} |C_m^n|$*. Similarly, the* depth *of the family is the mapping that associates* $n$ *to the supremum of the depths of each* $C_m^n$*. The class* $\mathrm{FAC}_v^0$*, standing for* functions in $\mathrm{AC}^0$ with variable output length*, is the class of functions expressible as a family of constant-depth, polysize circuits. The class* $\mathrm{FAC}_v^0(\mathcal{V})$ *is defined in the same fashion as* $\mathrm{AC}^0(\mathcal{V})$*, and we let* $\mathrm{FACC}_v^0 = \mathrm{FAC}_v^0(\mathrm{MOD})$.

*Remark 1.*

- Any function $\tau$ in $\mathrm{FAC}^0_{\mathrm{v}}(\mathcal{V})$ is such that $n \mapsto \max_{u \in \Sigma^n} |\tau(u)|$ has value in $\mathbb{N}$, that is, for a given input size, there is a finite number of possible output sizes. More precisely, this mapping is polynomially bounded. We show this implies that $\tau$ is $\mathrm{AC}^0(\mathcal{V})$-continuous. Let $(C^n_m)_{n,m\geq 0}$ be the circuit family for $\tau$. Given a language $L$ in $\mathrm{AC}^0(\mathcal{V})$ expressed by the circuit family $(D_n)_{n>0}$, $\tau^{-1}(L) \cap \Sigma^n$ is recognized by the circuit that applies a polynomial number of circuits $C^n_m$ to the input, and checks that the only $m$ such that $C^n_m$ outputs $(v, 1)$ is such that $v \in D_m$.
- If for any $n$ there is an $m$ such that $\tau(\Sigma^n) \subseteq \Sigma^m$, i.e., if $\tau$ is not of variable output length, then $\tau \in \mathrm{FAC}^0_{\mathrm{v}}(\mathcal{V})$ is equivalent to $\tau \in \mathrm{FAC}^0(\mathcal{V})$.
- We will be interested in functions from $\Sigma^*$ to $\mathbb{N}$, and will speak of their circuit definability. In this context, the function is either seen as taking value in $\{1\}^*$, and dealt with using a variable-output-length circuit, or taking value in $\{0, 1\}^*$ using an $\mathrm{FAC}^0$-like circuit, the output value then corresponding to the position of the last 1 in the output. These two views are equivalent, and hence we do not rely on a specific one. We note that (general) transductions from $\Sigma^*$ to $\{1\}^*$ have been extensively studied in [7]; therein, Choffrut and Schützenberger show that such a function is a transduction iff it has a strong form of *uniform* continuity, akin to the one presented in the introduction, with longest common *subwords* instead of prefixes.

## 3 Separability, Definability, and Lm-Varieties of Stamps

Recall that the transition monoid of an automaton $A$ is the monoid under composition consisting of the functions $f_w \colon Q \to Q$ defined by $f_w(q) = \delta(q, w)$. Historically, regular languages were studied through properties of the transition monoids of their minimal automata (the so-called *syntactic monoids*). As previously mentioned, the minimal automata for $\mathrm{EVEN} \in \mathrm{AC}^0$ and $\mathrm{PARITY} \notin \mathrm{AC}^0$ have the same transition monoid, hence the class $\mathrm{AC}^0 \cap \mathrm{REG}$ admits no syntactic monoid characterization. Starting with [2], the interest shifted to *transition morphisms* of automata, i.e., the surjective morphisms $\varphi \colon w \mapsto f_w$. It is indeed shown therein that a regular language is in $\mathrm{AC}^0$ iff $\varphi(\Sigma^s) \cup \{\varphi(\varepsilon)\}$ is an aperiodic monoid for some $s > 0$ and $\varphi$ associated with the minimal automaton.

A *stamp* is a surjective morphism from a free monoid to a finite monoid. A systematic study of the classes of languages described by stamps turned out to be a particularly fruitful research endeavor of the past decade [6,9,15,18]. Our use of this theory will however be kept minimal, and we will strive to only appeal to it in this section. The goal of the forthcoming Lemma 1 is indeed to express algebraic properties in a language-theoretic framework only.

Given a stamp $\varphi \colon \Sigma^* \to M$, we say that $L$ is *recognized* by $\varphi$ if there is a set $E \subseteq M$ such that $L = \varphi^{-1}(E)$—in this case, we also say that $L$ is recognized by $M$, which corresponds to the usual definition of recognition (e.g., [17]). We say that a stamp $\varphi \colon \Sigma^* \to M$ *lm-divides* a stamp $\psi \colon T^* \to N$ if $\varphi = \eta \circ \psi \circ h$, where $h \colon \Sigma^* \to T^*$ is an lm-morphism and $\eta \colon N \to M$ is a partial surjective

morphism. The *product* of two stamps $\varphi$ and $\psi$ with the same domain $\Sigma^*$ is the stamp mapping $a \in \Sigma$ to $(\varphi(a), \psi(a))$. Finally, an lm-variety of stamps is a class of stamps containing the stamps $\Sigma^* \to \{1\}$ and closed under lm-division and product. An Eilenberg theorem holds for lm-varieties: there is a one-to-one correspondence between lm-varieties of stamps and the lm-varieties of languages they recognize [18]. We show:

**Lemma 1.** *Let $A$ be an automaton, $\mathbf{V}$ an lm-variety of stamps, and $\mathcal{V}$ its corresponding lm-variety of languages. The following are equivalent:*

*(i) The transition morphism of $A$ is in $\mathbf{V}$;*
*(ii) $A$ is $\mathcal{V}$-all-definable;*
*(iii) $A$ is $\mathcal{V}$-all-separable.*

*Proof. (i) → (ii).* Let $\varphi$ be the transition morphism of $A$. Then $L(A, q) = \varphi^{-1}(E)$ where $E = \{f_w \mid f_w(q_0) = q\}$, hence $L(A, q) \in \mathcal{V}$.
*(ii) → (iii).* This is immediate, as $L(A, q)$ separates $q$ from any other state.
*(iii) → (i).* Write $L_{q,q'}$ for the language separating $q$ from $q'$. As each of these are recognized by stamps in $\mathbf{V}$ and $\mathbf{V}$ is closed under product, the language $L_q = \cap_{q' \neq q} L_{q,q'}$ is also recognized by a stamp in $\mathbf{V}$. Similarly, taking the product of the stamps recognizing the different $L_q$'s, we see that all of the $L_q$'s are recognized by the same stamp $\psi\colon \Sigma^* \to N$ in $\mathbf{V}$; let thus $E_q$ be such that $L_q = \psi^{-1}(E_q)$. Let $\varphi\colon \Sigma^* \to M$ be the transition morphism of $A$. We claim that $\varphi$ lm-divides $\psi$, concluding the proof as $\mathbf{V}$ is closed under lm-division. Define $\eta\colon N \to M$ by $\eta(\psi(w)) = \varphi(w)$. If $\eta$ is well-defined, then it is a surjective morphism, and we are done as $\varphi = \eta \circ \psi$. Suppose $\varphi(u) \neq \varphi(v)$, then there is a $p \in Q$ such that $\delta(p, u) = q$ and $\delta(p, v)$ is either undefined or a state $q' \neq q$. Let $w$ be a word such that $\delta(q_0, w) = p$, then $\psi(wu) \in E_q$ and $\psi(wv) \notin E_q$, hence $\psi(u) \neq \psi(v)$, showing that $\eta$ is well-defined. □

*Remark 2.* For $\mathcal{V} = \mathcal{QA}$ and $\mathcal{V} = \mathcal{M}_{\mathrm{sol}}$, the properties of Lemma 1 are decidable.

*As advertised, the rest of this paper will now be free from (lm-varieties of) stamps except for a brief incursion when discussing our results in Sect. 6. Lemma 1 enables a study that stands in the algebraic tradition with no appeal to its tools.*

## 4 The Transductions in $\mathbf{FAC}^0_{\mathbf{v}}(\mathcal{V})$

In sharp contrast with the work of Reutenauer and Schützenberger [16], we are especially interested in the shape of the outputs of the transduction. It turns out that most of its complexity is given by the following output-length function:

**Definition 2 ($\tau_{\#}$).** *Let $\tau$ be a transduction. The function $\tau_{\#}\colon \Sigma^* \to \mathbb{N}$ is the output-length function of $\mathsf{MinT}(\tau)$ with all the states deemed final. In symbols, $\tau_{\#}(w) = |\nu(q_0, w)|$, with $\mathsf{MinT}(\tau)$ as the underlying transducer.*

**Theorem 1.** *Let $\tau$ be a transduction and $\mathcal{V}$ be such that $\mathrm{AC}^0(\mathcal{V}) \cap \mathrm{REG} = \mathcal{V}$. The following constitutes a chain of implications:*

*(i)* $\tau \in \mathrm{FAC}^0_v(\mathcal{V})$*;*
*(ii)* $\tau$ *is* $\mathrm{AC}^0(\mathcal{V})$*-continuous;*
*(iii)* $\tau$ *is* $\mathcal{V}$*-continuous;*
*(iv)* $\mathsf{MinT}(\tau)$ *is* $\mathcal{V}$*-all-definable.*

*Moreover, if* $\tau_\# \in \mathrm{FAC}^0_v(\mathcal{V})$ *then (iv) implies (i). Somewhat conversely, (i) implies* $\tau_\# \in \mathrm{FAC}^0_v(\mathcal{V})$*.*

*Proof. (i)* $\to$ *(ii).* This was alluded to in Remark 1.
*(ii)* $\to$ *(iii).* This follows from the closure under inverse transductions of REG and the hypothesis that the regular languages of $\mathrm{AC}^0(\mathcal{V})$ are in $\mathcal{V}$.
*(iii)* $\to$ *(iv).* Let $q, q'$ be two states of $A = \mathsf{MinT}(\tau)$. We show that we can separate $q$ from $q'$. We distinguish the following cases, that span all the possibilities thanks to the output-minimality of $A$. In each case, we build a language $L$ separating $L(A, q)$ and $L(A, q')$, with $L \in \mathcal{V}$ relying on continuity and on $\mathcal{V}$ being an lm-variety by hypothesis. By Lemma 1, we then conclude *(iv)*.

- *Case 1:* There is a $w$ such that only one of $\delta(q, w)$ or $\delta(q', w)$ is in $F$. We suppose $\delta(q, w) \in F$, without loss of generality as $\mathcal{V}$ is closed under complement. Let $L = (\tau^{-1}(T^*))w^{-1}$, a word $x$ is in $L$ iff $\delta(q_0, xw) \in F$. This is the case for all words in $L(A, q)$ and for none in $L(A, q')$, hence $L$ separates these languages.
- *Case 2:* There is a $w$ such that both $\delta(q, w)$ and $\delta(q', w)$ are in $F$, and words $u, u'$ such that $\nu(q, w) = u \neq u' = \nu(q', w)$. Then we have two possibilities:
- *Case 2.1:* If $|u| = |u'|$. For a word $x \in \Sigma^*$, if $\tau(xw)$ ends with $u$, then $\delta(q_0, x)$ cannot be $q'$. Hence $L = (\tau^{-1}(T^*u))w^{-1} \in \mathcal{V}$ separates $L(A, q)$ from $L(A, q')$.
- *Case 2.2:* If $|u| \neq |u'|$. Define $k \in \mathbb{N}$ to be such that $|u| \not\equiv |u'| \mod k$, and let $s \in \{0, 1, \ldots, k-1\}$. Let $x \in \Sigma^*$ be such that $s \equiv |\nu(q_0, x)| \mod k$. Then if $\delta(q_0, x) = q$, we have $|\tau(xw)| \equiv s + |u| \mod k$, while $\delta(q_0, x) = q'$ makes this equation false. Hence the union $L$ over every $s$ of the languages $(\tau^{-1}(T^{k\mathbb{N}+s+|u|}))w^{-1}$ separates $L(A, q)$ from $L(A, q')$.

*(iv)* $\to$ *(i), assuming* $\tau_\# \in \mathrm{FAC}^0_v(\mathcal{V})$. We construct an $\mathrm{FAC}^0_v(\mathcal{V})$ circuit family for $\tau$. Fix an input size $n$ and an output size $m$. Given an input $x = x_1 x_2 \cdots x_n$, we first check, using $\tau_\#$, that the output length of $\tau$ on $x$ is indeed $m$, and wire this answer properly to the $(m+1)$-th output bit. Next, the $j$-th output bit, $1 \leq j \leq m$, is computed as follows. We apply $\tau_\#$ to every prefix of $x$, until we find an $i$ such that $\tau_\#(x_{<i}) < j \leq \tau_\#(x_{\leq i})$, where $x_{<i} = x_1 x_2 \cdots x_{i-1}$ and similarly for $x_{\leq i}$. Relying on the languages $L(A, q)$, we find the state $q$ in $\mathsf{MinT}(\tau)$ reached by $x_{<i}$, and let $u = \nu(q, x_i)$. The $j$-th output bit then corresponds to the $(j - \tau_\#(x_{<i}))$-th letter of $u$.

*(i)* $\to$ $(\tau_\# \in \mathrm{FAC}^0_v(\mathcal{V}))$. Suppose *(i)*, this implies *(iv)*. We construct an $\mathrm{FAC}^0(\mathcal{V})$ circuit family for $\tau_\#$. Fix the input size $n$, and let $x = x_1 x_2 \cdots x_n$ be the input. We can check, using the languages $L(\mathsf{MinT}(\tau), q)$, in which state $q$ the word $x$ ends when read. Let $w_q$ be a fixed word such that $\delta(q, w_q) \in F$, and let $r = |\nu(q, w_q)|$. It suffices now to plug the word $xw_q$ in the circuit for $\tau$; the value of $\tau_\#(x)$ is then the length of $\tau(xw_q)$ minus $r$. □

*Remark 3.* The proof of Theorem 1 shows that $\mathsf{MinT}(\tau)$, and hence $\tau_{\#}$, can be arbitrarily chosen as long as it is output-minimal. The role of $\tau_{\#}$ is discussed at greater length in Sect. 6.

## 5 An Application to $\mathrm{AC}^0$ and $\mathrm{ACC}^0$

Our primary focus is on the *decidability* of the membership of transductions in small-complexity classes. Theorem 1, while providing a characterization of these transductions, does not come with a decidable property in the general case—even when some conjectured separations are presupposed. With $\mathrm{AC}^0$ and $\mathrm{ACC}^0$, the functions $\tau_{\#}$ that can be expressed with circuits can however be characterized.

**Definition 3 (Constant Ratio).** *A transducer has* constant ratio *if every two words of the same length looping on a state produce outputs of the same length from this state. In symbols, for any state $q$ and any words $u, v$ of the same length, $\delta(q,u) = \delta(q,v) = q$ implies $|\nu(q,u)| = |\nu(q,v)|$.*

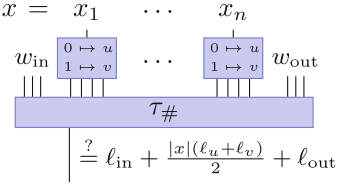
*Remark 4.* The name of the latter property stems from the fact that in such a transducer, for any state $q$, there is a ratio $\theta$ such that if $\delta(q,u) = q$, then $|\nu(q,u)| = \theta|u|$. Indeed, suppose a transducer has constant ratio, and let $u$ and $v$ be words with $\delta(q,u) = \delta(q,v) = q$ for some $q$. Write $|\nu(q,u)| = \theta_1|u|$ and $|\nu(q,v)| = \theta_2|v|$. Then $x = u^{|v|}$ and $y = v^{|u|}$ are of the same length, and $\theta_1|u| \times |v| = |\nu(q,x)| = |\nu(q,y)| = \theta_2|v| \times |u|$, hence $\theta_1 = \theta_2$.

**Lemma 2.** (Assuming $\mathrm{ACC}^0 \neq \mathrm{TC}^0$.) *Let $\tau$ be a transduction. If $\tau_{\#}$ is in $\mathrm{FACC}^0_v$, then $\mathsf{MinT}(\tau)$ has constant ratio.*

*Proof.* Suppose that $A = \mathsf{MinT}(\tau)$ does not have constant ratio. We give a circuit family in $\mathrm{AC}^0$ with $\tau_{\#}$-gates for the language $L = \{w \in \{0,1\}^* \mid |w|_0 = |w|_1\}$, which is complete for $\mathrm{TC}^0$. Hence $\tau_{\#}$ cannot admit an $\mathrm{FACC}^0_{\mathrm{v}}$ circuit family.

As $A$ does not have constant ratio, there are a state $q$ in $A$ and two words $u, v \in \Sigma^*$ of the same length, such that $\delta(q,u) = \delta(q,v) = q$ and $\ell_u = |\nu(q,u)|$ is different from $\ell_v = |\nu(q,v)|$. Further, let $w_{\mathrm{in}}$ (resp. $w_{\mathrm{out}}$) be such that $\delta(q_0, w_{\mathrm{in}}) = q$ (resp. $\delta(q, w_{\mathrm{out}}) \in F$), and let $\ell_{\mathrm{in}} = |\nu(q_0, w_{\mathrm{in}})|$ (resp. $\ell_{\mathrm{out}} = |\nu(q, w_{\mathrm{out}})|$).

We describe the circuit for $L$ for input size $n$. Let $x$ denote the input. First, the circuit transforms each 0 into $u$, and each 1 into input. The circuit can be graphically represented as follows:




First, the circuit transforms each 0 into $u$, and each 1 into $v$—this can be done as $|u| = |v|$. Then $w_{\mathrm{in}}$ is prepended and $w_{\mathrm{out}}$ appended to it, and the resulting word $x'$ is fed to $\tau_{\#}$. The output is $\ell_{\mathrm{in}} + |x|_0 \times \ell_u + |x|_1 \times \ell_v + \ell_{\mathrm{out}}$, that is:

$$\tau_{\#}(x') = \ell_{\mathrm{in}} + \frac{1}{2}(|x|(\ell_u + \ell_v) + (|x|_0 - |x|_1)(\ell_u - \ell_v)) + \ell_{\mathrm{out}} \ .$$

Now $(|x|_0 - |x|_1)(\ell_u - \ell_v)$ cancels out iff $x$ has as many 0's as 1's. Hence $x \in L$ iff the output of $\tau_\#$ is $\ell_{\text{in}} + \frac{1}{2}|x|(\ell_u + \ell_v) + \ell_{\text{out}}$, which is verifiable in $\text{AC}^0$. □

Having a constant ratio provides an easy way to compute the length function:

**Lemma 3.** *Let $\tau$ be a transduction. If $\mathsf{MinT}(\tau)$ has constant ratio and is $\mathcal{V}$-all-definable, then $\tau_\#$ is in $\text{FAC}^0_v(\mathcal{V})$.*

*Proof.* The circuit for $\tau_\#$ guesses a path without cycles for the input word, and checks that, modulo cycling, it is indeed a correct path for it. The output value is then entirely determined by the positions of the input at which this underlying path is taken. We now give a more precise construction.

Let $\pi = (q_0, a_0)(q_1, a_1)\cdots(q_k, a_k) \in (Q \times \Sigma)^*$ be an *accepting simple path* in $A = \mathsf{MinT}(\tau)$, that is, $q_{i+1} = \delta(q_i, a_i)$, $q_{k+1} = \delta(q_k, a_k) \in F$, and for $i \neq j$, $q_i \neq q_j$. There is a finite number of such paths, and for each of them, the circuit contains the following subcircuit. The subcircuit checks that the path in $A$ for the input word follows $\pi$, that is, if all the cycles are removed, then the resulting path is $\pi$. To do so, for each possible values of $1 \leq p_0 < p_1 < \cdots < p_k \leq n$ such that $\sum p_i = n$ (there is a polynomial number of them), the subcircuit checks for all $i \leq k$ that the prefix of length $p_i - 1$ of the input is in $L(A, q_i)$, and that the input at position $p_i$ is $a_i$. If this holds for all $i$, then the input word follows the path $\pi$, possibly cycling on each of the states $q_i$, $i \leq k+1$, and the output length is entirely determined. Indeed, with $\theta_0, \theta_1, \ldots, \theta_{k+1}$ the ratios of $q_0, q_1, \ldots, q_{k+1}$ respectively, and $\ell$ the sum of the output lengths of the transitions in $\pi$ (i.e., $\tau_\#(a_0 a_1 \cdots a_k)$), the value of $\tau_\#$ on the input is:

$$\theta_0 \times (p_0 - 1) + \textstyle\sum_{i=1}^{k} \theta_i \times (p_i - p_{i-1} - 1) + \theta_{k+1} \times (n - p_k) + \ell.$$ □

**Corollary 1.** *Let $\tau$ be a transduction. The following are equivalent, where the "resp." part assumes $\text{ACC}^0 \neq \text{TC}^0$:*

*(i) $\tau \in \text{FAC}^0_v$ (resp. $\in \text{FACC}^0_v$);*
*(ii) $\tau$ is continuous for $\text{AC}^0$ (resp. for $\text{ACC}^0$) and $\mathsf{MinT}(\tau)$ has constant ratio;*
*(iii) $\tau$ is continuous for $\mathcal{QA}$ (resp. for $\mathcal{M}_{sol}$) and $\mathsf{MinT}(\tau)$ has constant ratio;*
*(iv) $\mathsf{MinT}(\tau)$ is all-definable for $\mathcal{QA}$ (resp. for $\mathcal{M}_{sol}$) and has constant ratio.*

*Remark 5.* It should be noted that the choice of $\mathsf{MinT}(\tau)$ is again irrelevant. Either all the output-minimal transducers for $\tau$ are constant ratio, or none are.

**Theorem 2.** *It is decidable whether a transducer realizes an $\text{FAC}^0_v$ function. If it does, then a circuit family can be constructed. The same holds for $\text{FACC}^0_v$ assuming $\text{ACC}^0 \neq \text{TC}^0$.*

*Proof.* This is a direct consequence of Corollary 1, together with the minimization algorithm of [8], the fact that $\mathcal{V}$-all-definability is decidable, and the fact that it can be checked that a transducer has constant ratio: it is indeed enough to check the property on cycles that do not go twice in the same state except for the first. The constructions of Theorem 1 and Lemma 3 are then effective. □

Corollary 1 can be slightly strengthened for $\text{AC}^0$, as in this case:

**Proposition 3.** *If $\tau$ is $\mathcal{QA}$-continuous, then $\mathsf{MinT}(\tau)$ has constant ratio.*

*Proof.* This is a variant of Lemma 2, where we only rely on the inverse image of $\tau$ instead of a full circuit construction.

Suppose that $A = \mathsf{MinT}(\tau)$ does not have constant ratio. There are a state $q$ in $A = \mathsf{MinT}(\tau)$ and two words $u, v \in \Sigma^*$ of the same length, such that reading $u$ (resp. $v$) from $q$ produces an output of length $\ell_u$ (resp. $\ell_v$), and $\ell_u < \ell_v$. Now the words $y = u^{\ell_v}$ and $z = v^{2\ell_u}$ are such that $y$ produces an output of size $\ell_y = \ell_u \times \ell_v$, and $z$ produces an output of size $\ell_z = \ell_v \times 2\ell_u = 2\ell_y$.

Now if $\tau$ is a $\mathcal{QA}$-continuous transduction, so is the function $\tau'$ mapping $x \in \{0,1\}^*$ to a word on $\{a\}^*$ with $\ell_y \times |x|_1 + \ell_z \times |x|_0$ letters $a$—it is simply a matter of replacing 1 with $y$, 0 with $z$, and correctly reaching the state $q$. But $\tau'^{-1}(a^{2\ell_y\mathbb{N}})$ is PARITY: indeed, $x$ has an odd number of 1 iff $\tau'(x)$ contains an odd number of blocks $a^{\ell_y}$. Hence $\tau'$ is not $\mathcal{QA}$-continuous, and neither is $\tau$. □

## 6 Discussion and Limitations

1 We note that Proposition 3 fails in the case of $\mathrm{ACC}^0$, as the following example shows. Consider the morphism $h\colon a \mapsto a,\ b \mapsto aa$. As the regular languages of $\mathrm{ACC}^0$, $\mathcal{M}_{\mathrm{sol}}$, are closed under inverse morphism (this is a consequence of $\mathcal{M}_{\mathrm{sol}}$ being a variety), $h$ is $\mathcal{M}_{\mathrm{sol}}$-continuous. However, $\mathsf{MinT}(h)$ does not have constant ratio. This was already noted in a different setting by Lange and McKenzie [13].

2 The major role that $\tau_{\#}$ plays in Theorem 1 raises several questions. First, is it the case that all the complexity of a transduction is characterized by its length function? In symbols, is it true that $\tau_{\#} \in \mathrm{FAC}^0_{\mathrm{v}}(\mathcal{V}) \Rightarrow \tau \in \mathrm{FAC}^0_{\mathrm{v}}(\mathcal{V})$? The following example shows that it is not. Consider the transduction from $\{0,1\}^*$ to $\{a,b\}^*$ that outputs $a$ if the word read so far is in PARITY, and $b$ otherwise. Then $\tau_{\#}$ is total and maps every word to its length, it is thus in $\mathrm{FAC}^0_{\mathrm{v}}$. However, $w$ is in PARITY iff the last letter of $\tau(w)$ is an $a$, that is, $\tau^{-1}(\{a,b\}^*a)$ is PARITY, hence $\tau$ is not $\mathcal{QA}$-continuous, thus cannot be in $\mathrm{FAC}^0_{\mathrm{v}}$.

Next, going down two levels in the statement of Theorem 1, we may wonder whether the $\mathcal{V}$-continuity of $\tau$ is equivalent to that of $\tau_{\#}$. One direction is true, but its converse fails, as the previous example shows:

**Proposition 4.** *If $\tau$ is $\mathcal{V}$-continuous, then so is $\tau_{\#}$.*

*Proof.* Suppose $\tau$ is $\mathcal{V}$-continuous. Let $E$ be a set of integers, and write $\Sigma^E$ for the words of lengths in $E$. Suppose $\{a\}^E$ is in $\mathcal{V}$; we show $\tau_{\#}^{-1}(E) \in \mathcal{V}$.

From Theorem 1, $A = \mathsf{MinT}(\tau)$ is $\mathcal{V}$-all-definable. Let $q$ be a state of $A$, and $w$ a word mapping $q$ to a final state while outputing $u$. Then $(\tau^{-1}(\Sigma^E.u))w^{-1} \cap L(A,q)$ is in $\mathcal{V}$, as $\tau$ is $\mathcal{V}$-continuous and $\{a\}^E \in \mathcal{V}$. Now the union of all these sets for all states $q$ is precisely $\tau_{\#}^{-1}(E)$, hence it is in $\mathcal{V}$. □

3 Our interest in circuits obscured an equally interesting problem: characterizing the $\mathcal{V}$-continuous transductions. A general question raised by our characterization is:

*Question 1.* Which lm-varieties $\mathcal{V}$ verify the following statement? A transduction $\tau$ is $\mathcal{V}$-continuous iff $\mathsf{MinT}(\tau)$ is $\mathcal{V}$-all-definable and $\tau_{\#}$ is $\mathcal{V}$-continuous.

A direct consequence of Proposition 3 is that $\mathcal{V} = \mathcal{QA}$ verifies Question 1. Another such class is given in [16]; therein, Reutenauer and Schützenberger show that the property holds for $\mathcal{V} = \mathcal{G}$, the (lm-)variety of group languages, that is, languages with a group as syntactic monoid. More precisely, they show that $\tau$ is $\mathcal{G}$-continuous iff the transition monoid of $\mathsf{MinT}(\tau)$ is a group; this latter property is equivalent to: the transition morphism of $\mathsf{MinT}(\tau)$ is a stamp $\Sigma^* \to G$, for $G$ a group. The set of such stamps is an lm-variety of stamps (see [6]), thus by Lemma 1, their characterization is indeed of the form of Question 1.

4 It is interesting to note that the property on $\tau_{\#}$ of Question 1 vanishes for groups: this can be seen as a consequence of Reutenauer and Schützenberger's characterization itself, as $\tau_{\#}$ has the same transition monoid as $\mathsf{MinT}(\tau)$. On the other hand it is shown in the same article that there are transductions with an aperiodic monoid that are not continuous for aperiodic languages. This raises the question:

*Question 2.* Which lm-varieties $\mathcal{V}$ verify the following statement? If a transduction $\tau$ is such that $\mathsf{MinT}(\tau)$ is $\mathcal{V}$-all-definable, then $\tau_{\#}$ is $\mathcal{V}$-continuous.

5 Recall that a nondeterministic transduction is functional iff it is realized by an unambiguous transduction (see, e.g., [3]). As circuits can read the input multiple times and in any direction, it seems that they can handle deterministic and unambiguous transductions in the same fashion. Hence a generalization of Theorem 1 to the unrestricted case of functional transductions should hold.

**Acknowledgment.** We thank Michael Blondin, Michael Hahn, and the referees.

## References

1. Barrington, D.A.M.: Bounded-width polynomial size branching programs recognize exactly those languages in $NC^1$. J. Comp. Syst. Sc. **38**, 150–164 (1989)
2. Barrington, D.A.M., Compton, K., Straubing, H., Thérien, D.: Regular languages in $NC^1$. J. Comput. Syst. Sci. **44**(3), 478–499 (1992)
3. Berstel, J.: Transductions and Context-Free Languages, Leitfäden der Angewandten Mathematik und Mechanik LAMM. Teubner, Stuttgart (1979)
4. Beyersdorff, O., Datta, S., Krebs, A., Mahajan, M., Scharfenberger-Fabian, G., Sreenivasaiah, K., Thomas, M., Vollmer, H.: Verifying proofs in constant depth. TOCT **5**(1), 2 (2013)
5. Bojańczyk, M.: Transducers with Origin Information. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) ICALP 2014, Part II. LNCS, vol. 8573, pp. 26–37. Springer, Heidelberg (2014)
6. Chaubard, L., Pin, J.É., Straubing, H.: First-order formulas with modular predicates. In: LICS, pp. 211–220. IEEE (2006)

7. Choffrut, C., Schützenberger, M.P.: Counting with rational functions. Theor. Comput. Sci. **58**(1–3), 81–101 (1988)
8. Choffrut, C.: A generalization of Ginsburg and Rose's characterization of G-S-M mappings. In: Maurer, H.A. (ed.) ICALP 1979. LNCS, vol. 71, pp. 88–103. Springer, Heidelberg (1979)
9. Esik, Z., Ito, M.: Temporal logic with cyclic counting and the degree of aperiodicity of finite automata. Acta Cybern. **16**(1), 1–28 (2003)
10. Filiot, E., Krishna, S.N., Trivedi, A.: First-order definable string transformations. In: Raman, V., Suresh, S.P. (eds.) FSTTCS. LIPIcs, vol. 29, pp. 147–159 (2014)
11. Furst, M., Saxe, J.B., Sipser, M.: Parity, circuits, and the polynomial-time hierarchy. Theor. Comput. Syst. **17**, 13–27 (1984)
12. Koucký, M., Pudlák, P., Thérien, D.: Bounded-depth circuits: separating wires from gates. In: STOC, pp. 257–265. ACM (2005)
13. Lange, K.-J., McKenzie, P.: On the complexity of free monoid morphisms. In: Chwa, K.-Y., Ibarra, O.H. (eds.) ISAAC 1998. LNCS, vol. 1533, pp. 247–255. Springer, Heidelberg (1998)
14. Lautemann, C., McKenzie, P., Schwentick, T., Vollmer, H.: The descriptive complexity approach to LOGCFL. J. Comput. Syst. Sci. **62**(4), 629–652 (2001)
15. Pin, J.É., Straubing, H.: Some results on C-varieties. RAIRO-Theor. Inf. Appl. **39**(01), 239–262 (2005)
16. Reutenauer, C., Schützenberger, M.P.: Variétés et fonctions rationnelles. Theor. Comput. Sci. **145**(1–2), 229–240 (1995)
17. Straubing, H.: Finite Automata, Formal Logic, and Circuit Complexity. Birkhäuser, Boston (1994)
18. Straubing, H.: On logical descriptions of regular languages. In: Rajsbaum, S. (ed.) LATIN 2002. LNCS, vol. 2286, pp. 528–538. Springer, Heidelberg (2002)
19. Vollmer, H.: Introduction to Circuit Complexity. Springer-Verlag, Berlin (1999)