The Regular Languages of Wire Linear AC 0

Michaël Cadilhac, Charles Paperman

The Regular Languages of Wire Linear AC^0

Michaël Cadilhac
School of Computing, DePaul University, 243 S. Wabash Ave., Chicago, 60604, IL, USA

Charles Paperman
Univ. Lille, CNRS, INRIA, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France

Abstract

In this paper, the regular languages of wire linear AC^0 are characterized as the languages expressible in the two-variable fragment of first-order logic with regular predicates, $FO^2[reg]$. Additionally, they are characterized as the languages recognized by the algebraic class **QLDA**. The class is shown to be decidable and examples of languages in and outside of it are presented.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory

Keywords and phrases circuit complexity, regular languages, algebraic language theory

Digital Object Identifier 10.4230/LIPIcs...

Contents

1 Introduction **1**

2 Preliminaries **3**

3 Algebra, logic, and circuits **5**
3.1 From algebra to logic 5
3.2 From logic to circuits 7
3.3 Back to algebra 8
3.4 Closing the circle: from circuits to algebra 11

4 Applications **12**
4.1 Decidability 12
4.2 $LAC^0 \cap$ Reg, Straubing and Crane Beach properties 12
4.3 Bounded-depth Dyck languages 13

5 Conclusion **14**

1 Introduction

A recurring theme in the work of Klaus-Jörn Lange is the interplay of logic, algebra, and circuit complexity. In this paper dedicated to his 70th birthday, we exhibit one of these tight relationships by looking at the class of regular languages recognized by circuits of very low complexity.

For a family of Boolean circuits $(C_n)_{n\geq 0}$, where each C_i has i inputs and one output, its *language* is the set of words w such that $C_{|w|}$ outputs 1 when w is placed as input. Circuit complexity is the study of how certain parameters of C_n evolve as n grows. The classical class AC^0 is defined by taking Boolean circuits with unbounded fan-in, constant depth, and a polynomial number of gates. The regular languages of AC^0 are well understood: they have an algebraic characterization (the class **QA** of quasi-aperiodic stamps) and correspond to the logic FO[reg] of first-order sentences with regular predicates (see, e.g., the splendid account

of Straubing [17]). Our interest lies in restricting the number of gates and wires in these circuits even further and characterizing the regular languages that can still be defined.

Define LAC^0 as the subclass of AC^0 where the number of gates is restricted to be linear in the input length and $WLAC^0$ the subclass where the number of *wires* is restricted to be linear. Naturally, it is not useful to have more gates than wires, so $WLAC^0 \subseteq LAC^0$, this inclusion is in fact strict [10]. The class LAC^0 also enjoys a logical characterization: it is equivalent to $FO^2[arb]$, the two-variable fragment of first-order logic [10]. No such characterization is known for $WLAC^0$.

Perhaps surprisingly, there are still some important open problems in the regular language realms of LAC^0 and $WLAC^0$. Most striking among these:

▶ **Conjecture 1.** *Do* LAC^0 *and* $WLAC^0$ *recognize the same regular languages?*

In studying this question, Koucký, Pudlák, and Thérien [10] gave a beautiful characterization of a subclass of the regular languages in $WLAC^0$: those with a neutral letter. A language has a neutral letter c if c can be added or removed from words without impacting their membership to the language. Koucký et al. showed that, for neutral-letter languages, the regular languages of $WLAC^0$ are those definable in $FO^2[<]$. This raised the question of whether the full class of regular languages in $WLAC^0$ can be similarly characterized.

On the power of neutral letters. When adding a neutral letter to a language, a property on words as anodyne as *"each third position contains the letter a"* becomes *"each position such that the number of nonneutral letters before it is congruent to zero modulo three contains the letter a,"* which is much harder to implement (in fact, provably not in AC^0). Indeed, on inputs $w_1w_2\cdots w_6$ of length 6 and over the alphabet $\{a, b\}$, the former is implemented by a circuit that tests $w_3 = a \wedge w_6 = a$, while the latter, with c a neutral letter, is much more complex:

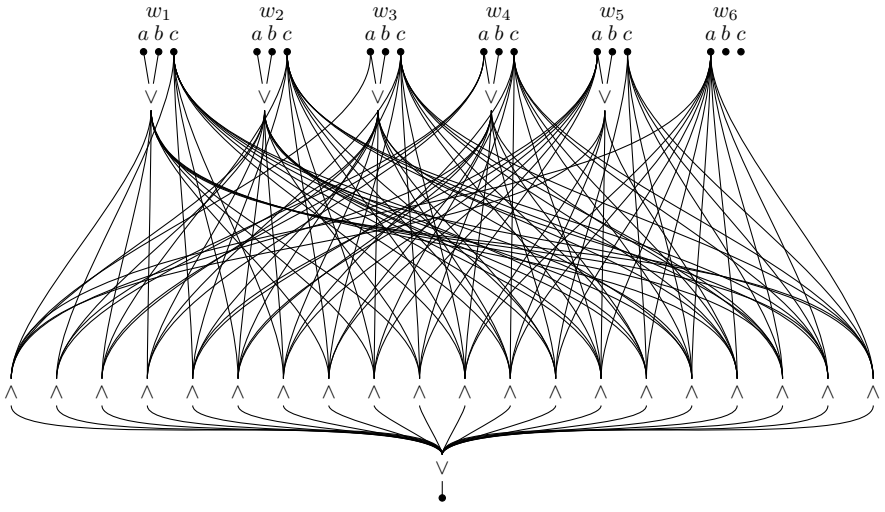

Figure 1 A circuit checking that each third nonneutral letter is a.

This brought the idea that, when considering only languages with a neutral letter, no interesting information can be extracted from the position numbers (besides order). In other words, the subset of languages with a neutral letter in a class ought to be much less complex than the whole class. This is formalized as the *Crane Beach property*, the name stemming

for the disproved conjecture of the same name which stated that the neutral-letter languages of FO[arb] were in FO[<]. For FO^2[arb], this is open:

▶ **Conjecture 2** (Crane Beach property). *Are all the neutral-letter languages of* FO^2[arb] *definable in* FO^2[<]*?*

Circling back to WLAC^0, Koucký et al. showed that all the regular languages in WLAC^0 with a neutral-letter belong to FO^2[<]. This left the possibility that the whole class of regular languages of WLAC^0 could necessitate more expressive power than what FO^2[<] offers.

Contributions. We show that $\mathrm{WLAC}^0 \cap \mathrm{Reg}$ is precisely FO^2[reg], or in other words, that dropping the neutral-letter restriction requires adding the regular predicates to the logic. We also provide an algebraic characterization of that class as **QLDA**. As this latter class is decidable, this leads to a decision procedure to test if a regular language belongs to WLAC^0. This comes short of providing an answer to the above open problems, but this shows in particular that Section 1 admits a positive answer iff FO^2[arb] has the so-called *Straubing property*:

▶ **Conjecture 3** (Straubing property). *Are all the regular languages of* FO^2[arb] *definable in* FO^2[reg]*?*

2 Preliminaries

We assume familiarity with regular languages, logic, and circuits, although we strive to keep this presentation self-contained. We write Reg for the class of regular languages.

Monoid, morphisms, quotient. A monoid is a set equipped with a binary associative operation, denoted multiplicatively, with a unit element. Within a finite monoid M, we write ω for the smallest value such that $a^\omega = a^{2\omega}$ for all $a \in M$. For an alphabet A, the set A^* is the free monoid generated by A, its unit element being the empty word ε. A morphism is a map $\phi\colon M \to N$ satisfying $\phi(ab) = \phi(a)\phi(b)$ and $\phi(1) = 1$, with $a, b \in M$ and 1 denoting the unit element of M and N. If for a morphism $\phi\colon A^* \to B^*$ there is a k such that $\phi(A) \subseteq B^k$, we call ϕ an *lm-morphism*, where lm stands for *length-multiplying*. Given a language L and a letter a, the *left quotient* of L by a is the set $a^{-1}L = \{v \mid av \in L\}$. The *right quotient* La^{-1} is defined symmetrically. An *lm-variety of languages* is a set of languages closed under the Boolean operations, quotient, and inverse lm-morphisms. We say that a language L has a neutral letter if there is a letter c such that $u \cdot c \cdot v \in L \Leftrightarrow u \cdot v \in L$ for all words u, v.

Logic. We work with first-order logics recognizing languages. For instance, the formula over the alphabet $\{a, b\}$

$$(\forall x)(\exists y)[\mathbf{max}(x) \vee (y = x + 1) \wedge (a(x) \leftrightarrow b(y))]$$

asserts that, in a given word w, for every position x there is another position y such that either x is the last position of the word (**max**) or y is just after x and w has different letters at x and y. The predicates **max** and $+1$ are *numerical predicates*, i.e., they only speak about the numerical positions, not the content of the input word. The predicates $a(\cdot)$ and $b(\cdot)$ are the *letter predicates*.

In this work, logics are specified by restricting two aspects:

- The number of variables; we define FO^2 as the logic with only two variables that can be reused (e.g., $(\exists x)(\forall y)[\cdots \wedge (\exists x)[\cdots]]$ is allowed). The unrestricted logic is simply written FO.
- The *numerical predicates* allowed. We will be using in particular $<$, $+1$ (the successor predicates, including adding any constant), and the modulo predicates asserting that a position is congruent to zero modulo a fixed number. These form the *regular predicates* and we write reg for that class (e.g., $FO^2[\text{reg}]$). The term "regular" stems from the fact that these are the properties on numerical positions that automata can express. More generally, we specify the allowed numerical predicates in bracket, e.g., $FO[<]$ or $FO^2[<, +1]$. If we allow any arbitrary numerical predicate, we write arb in the brackets; for instance, in FO[arb], a formula can use the predicate "these two positions are coprime."

We write $w \models \phi$, for a formula ϕ, to indicate that w satisfies ϕ, with the obvious meaning where quantifications are made over the positions of w. The language of ϕ is the set of words that satisfy it. A class of formulas is seen as the class of languages the formulas recognize.

Circuits. We denote by AC^0 the class of languages computed by families of constant-depth, polynomial-size circuits consisting of unbounded fan-in $\wedge$- and $\vee$-gates and unary $\neg$-gates. Such a family is an infinite set $(C_n)_{n\geq 0}$ where the circuit C_i has i inputs and one output gate; a (binary) word w is deemed accepted if the circuit $C_{|w|}$ outputs 1 when w is placed as input.

We will explore languages over alphabets larger than the binary alphabet; for these, an encoding of the alphabet is usually necessary. Our results are not influenced by the choice of encoding so we will not specify one.

We denote by LAC^0 the subclass of AC^0 languages that are computable by AC^0-circuits having a linear number of *gates*. We write $WLAC^0$ for the class with the added restriction of having a linear number of *wires*.

Stamps and varieties thereof.[1]

A stamp [16] is a surjective monoid morphism $\mu\colon A^* \to M$ from a free monoid to a finite monoid. A language $L \subseteq A^*$ is *recognized* by μ if there is a set $E \subseteq M$ such that $L = \mu^{-1}(E)$. The following two definitions are necessary for completeness but will not play an important role in what follows; if new to the reader, they can be safely skipped on first reading. We say that a stamp $\mu\colon A^* \to M$ lm-divides a stamp $\rho\colon B^* \to N$ if $\rho = \tau \circ \mu \circ \phi$ where $h\colon A^* \to B^*$ is an lm-morphism and $\tau\colon N \to M$ is a partial surjective morphism. The *product* of two stamps μ and ρ with the same domain A^* is the stamp mapping $a \in A$ to $(\mu(a), \rho(a))$.

Finally, an lm-variety of stamps is a class of stamps containing the stamps $A^* \to \{1\}$ and closed under lm-division and product. One salient feature of lm-varieties of stamps is that they correspond one-to-one to lm-varieties of languages: if $\mathbf{V}$ is an lm-variety of stamps, we write $\mathcal{V}$ for the lm-variety of languages that are recognized by the stamps in $\mathbf{V}$.

For a stamp $\mu\colon A^* \to M$, we write ω for the value of ω in M, so that for any word u, $\mu(u^\omega) = \mu(u^{2\omega})$. (In more details: $\mu(u^\omega) = \mu(u)^\omega$ since μ is a morphism, $\mu(u)^\omega = \mu(u)^{2\omega}$ by definition of ω, and $\mu(u)^{2\omega} = \mu(u^{2\omega})$ since μ is a morphism.)

The L operator: Augmenting varieties with local predicates. Let $\mu\colon A^* \to M$

[1] For the reader familiar with algebraic language theory: note that to simplify presentation, we do not define varieties of monoids and see them as varieties of stamps.

be a stamp. We are interested in looking at the portion of μ that does not care about neighboring information on positions. To do so, consider a nonempty word x such that $\mu(x)$ is an idempotent. If x appears in another word, say $w = uxv$, then x can be repeated any number of times without changing its image via μ, e.g., $\mu(w) = \mu(ux^{42}v)$. This means that the neighboring information between u and v was not important: this is the "portion of μ" we are interested in.

Formally, the *local stamps* of μ are derived from any idempotent in $e \in \mu(A^+)$ by:

$$\mu_e \colon \quad A^* \to e \cdot M \cdot e \qquad \text{(Note that } eMe \text{ is a monoid.)}$$
$$a \in A \mapsto e \cdot \mu(a) \cdot e$$

For a variety of stamps $\mathbf{V}$, we define $\mathbf{LV}$ as the variety of stamps whose local stamps are all in $\mathbf{V}$. It can be shown [15] that when $\mathbf{V}$ satisfies some technical property, one can see $\mathbf{LV}$ as $\mathbf{V}$ augmented with the neighboring information on positions.

The Q operator: Augmenting varieties with modulo predicates.

Let $\mu \colon A^* \to M$ be a stamp. Since $\mu(A)$ can be seen as an element of the powerset monoid of M, there is an integer $s > 0$ such that $\mu(A)^s = (\mu(A)^s)^2$; the smallest such s is the *stability index* of μ. By construction, $\mu(A)^s$ is closed under multiplication and we write $S(\mu)$ for that set with possibly an identity element added so that it is a monoid. The *stable stamp* of μ is that same stamp but taking A^s as alphabet, i.e.:

$$\rho \colon (A^s)^* \to S(\mu)$$
$$w \mapsto \mu(w).$$

For a variety of stamps $\mathbf{V}$, we define $\mathbf{QV}$ as the variety of stamps whose stable stamps belongs to $\mathbf{V}$. Intuitively, the stable stamp of μ embeds in the alphabet itself the information of modulo values of positions in the word. Again, it can be shown [15] that under some hypothesis on $\mathbf{V}$, one can see $\mathbf{QV}$ as $\mathbf{V}$ augmented with modulo counting on the positions.

Some lm-varieties. We will rely on a few classical lm-varieties of stamps:

- $\mathbf{A}$: the aperiodic stamps. These are the stamps μ that satisfy, for all words u, $\mu(u^\omega) = \mu(u^{\omega+1})$. It holds that $\mathcal{A} = \mathrm{FO}[<]$.
- $\mathbf{DA}$. Stamps therein are those that satisfy, for all words x, y, $\mu((xy)^\omega x (xy)^\omega) = \mu((xy)^\omega)$. The class $\mathcal{DA}$ enjoys a wealth of different characterizations [18], it holds in particular that $\mathcal{DA} = \mathrm{FO}^2[<]$.
- $\mathbf{QA}$. This is $\mathbf{Q}$ applied to $\mathbf{A}$ and was historically one of the first lm-varieties of stamps studied [2]. It holds that $\mathcal{QA} = \mathrm{FO}[\mathrm{reg}] = \mathrm{AC}^0 \cap \mathrm{Reg}$.

3 Algebra, logic, and circuits

As it is often witnessed in the low-level reaches of circuit complexity, we will observe that the class $\mathrm{WLAC}^0 \cap \mathrm{Reg}$ exhibits a beautiful interplay between algebra, logic, and circuits. With $\mathcal{NL}$ the class of languages with a neutral letter, Koucký, Pudlák, and Thérien [10] already showed:

▶ **Theorem 4** ([19, 10]). $\mathcal{DA} \cap \mathcal{NL} = \mathrm{FO}^2[<] \cap \mathcal{NL} = \mathrm{WLAC}^0 \cap \mathrm{Reg} \cap \mathcal{NL}$.

Our goal in this section is to fully characterize $\mathrm{WLAC}^0 \cap \mathrm{Reg}$ in a similar way. We prove the inclusion of the algebra-flavored class into the logical one in Section 3.1, then of the logic-flavored into WLAC^0 in Section 3.2. To close the chain of inclusions, we take a short detour in the purely algebraic world in Section 3.3 and conclude in Section 3.4.

3.1 From algebra to logic

The algebraic class we will be studying is **QLDA**: this is to be read as $\mathbf{Q}(\mathbf{L}(\mathbf{DA}))$, i.e., **DA** on which the **L** operator was applied, then the **Q** operator. Recall that $\mathcal{DA} = \mathrm{FO}^2[<]$. As we hinted, the **L** operator intuitively adds the local predicates and the **Q** operator the modulo predicates, it is thus not surprising that we will end up showing $\mathcal{QLDA} = \mathrm{FO}^2[\mathrm{reg}]$. This result was announced without proof in [7] and we provide a proof here through the chain of inclusions that will be concluded in Section 3.4. We start with:

▶ **Lemma 5.** $\mathcal{QLDA} \subseteq \mathrm{FO}^2[\mathrm{reg}]$.

Proof. We rely on the characterization $\mathcal{LDA} = \mathrm{FO}^2[<, +1]$ from [19, 1] (see also [13]) and carefully adapt the proof of $\mathcal{QA} \subseteq \mathrm{FO}[\mathrm{reg}]$ from [2].

Let $L \in \mathcal{QLDA}$ be recognized by a stamp $\mu\colon A^* \to M$ in **QLDA**, i.e., $L = \mu^{-1}(E)$ for some $E \subseteq M$. Let s be the stability index of μ, so that $S(\mu) = \mu(A)^s = \mu(A)^{2s}$. We have that:

$$L = \bigcup_{\substack{w \in A^*: \\ 0 \leq |w| < s}} L_w \cdot w,$$

where

$$L_w = \{u \in (A^s)^* \mid u \cdot w \in L\}.$$

To show that $L \in \mathrm{FO}^2[\mathrm{reg}]$, it is thus enough to show that each $L_w \cdot w$ is in $\mathrm{FO}^2[\mathrm{reg}]$, since the latter class is closed under union.

We first show that $L_w \in \mathrm{FO}^2[\mathrm{reg}]$ implies that $L_w \cdot w \in \mathrm{FO}^2[\mathrm{reg}]$. Let $w = a_1 a_2 \cdots a_r$ and $\phi \in \mathrm{FO}^2[\mathrm{reg}]$ recognize L_w. Let $\mathbf{max}_{\leftarrow r}(x)$ be the unary predicate that is true of a position x if x is not within r positions of the last position; note that $\mathbf{max}_{\leftarrow r}(x)$ can be expressed in $\mathrm{FO}^2[\mathrm{reg}]$. The formula ϕ can thus be relativized to $\mathbf{max}_{\leftarrow r}$; that is, every position quantified should satisfy $\mathbf{max}_{\leftarrow r}$. Let ϕ' be that formula; it recognizes the language $L_w \cdot (A^r)$. Now let $\mathbf{max}_i(x)$ be the predicate that is true of a position x if x is i positions away from the last position, this is again expressible in $\mathrm{FO}^2[\mathrm{reg}]$. The formula

$$\psi \equiv \bigwedge_{i=1,\ldots,r} (\exists x)[\mathbf{max}_{r-i}(x) \wedge a_i(x)]$$

recognizes the language $A^* \cdot w$. Hence $(\phi' \wedge \psi) \in \mathrm{FO}^2[\mathrm{reg}]$ recognizes $L_w \cdot w$.

Finally, we show that for any w, $L_w \in \mathrm{FO}^2[\mathrm{reg}]$. Let ρ be the stable stamp of μ. Consider $B = A^s$ as an alphabet, so that $\rho\colon B^* \to S(\mu)$. We start by viewing L_w as a language over B. A word $v \in B^*$ belongs to L_w iff

$$\rho(v) \in \underbrace{\{m \in S(\mu) \mid m \cdot \mu(w) \in E\}}_{F},$$

in other words, $L_w = \rho^{-1}(F)$, implying that ρ recognizes L_w. Consequently, since $\rho \in \mathbf{LDA}$ by definition of the **Q** operator, we have that $L_w \in \mathcal{LDA}$ as a language over B. As $\mathcal{LDA} = \mathrm{FO}^2[<, +1]$, we have a formula $\phi \in \mathrm{FO}^2[<, +1]$ recognizing L_w, again as a language over B.

We now modify ϕ so that it recognizes L_w over A. To do so, we first relativize each quantification so that only positions that are zero modulo s are considered; since the latter

predicate is regular, this results in a formula in $\mathrm{FO}^2[\mathrm{reg}]$. Next, since ϕ works over B, letter predicates in ϕ are of the form

$$a_1 a_2 \cdots a_s(x)$$

with each $a_i \in A$. This can be rewritten over A as

$$\bigwedge_{i=1,\dots,s} (\exists y)[y = x + i \wedge a_i(y)],$$

still resulting in a $\mathrm{FO}^2[\mathrm{reg}]$ formula. Finally, we can check that the length of the input word is zero modulo s using $(\exists x)[\mathbf{max}_0(x) \wedge (x \equiv 0 \mod s)]$. The resulting $\mathrm{FO}^2[\mathrm{reg}]$ formula defines L_w. ◀

3.2 From logic to circuits

Results showing that some logic classes correspond to some circuit classes abound in the literature, in particular in the corpus of Lange [11, 8, 3, 4, 12, 14]. The usual pattern is that the quantifier part (say, $\mathrm{FO}, \mathrm{FO}^2$, or more exotic logics with majority quantifiers) corresponds to the allowed circuit gates, while the numerical predicates (say, $+1, \mathrm{reg}, \mathrm{arb}$, or multiplication) correspond to the allowed computing power to wire the circuit, the so-called *uniformity* of the circuit class.

Recall that $\mathrm{FO}^2[\mathrm{arb}] = \mathrm{LAC}^0$, so if we change arb to reg, we expect a reduction in how intricate the gate wiring can be on the circuit side. In fact, our ability is so restricted that we cannot define more than a linear number of wires:

▶ **Lemma 6.** $\mathrm{FO}^2[\mathrm{reg}] \subseteq \mathrm{WLAC}^0$.

Proof. Let $\phi(x) \in \mathrm{FO}^2[\mathrm{reg}]$ — this denotes that ϕ *may* have one free variable, and that it is x. The proof is done by induction on the structure of ϕ and follows the steps in the proof of [9, Theorem 2]. We will build a circuit C for $\phi(x)$ that has n inputs and n outputs (the output gates need not all be distinct). The circuit C satisfies for every word w of length n: $w \models \phi(i)$ iff the i-th output of $C(w)$ is 1. We write $C^{(i)}(w)$ for that output and simply $C^{(i)}$ for the output gate. We will make sure that the number of wires of C is linear in n.

Without loss of generality, we assume that ϕ does not contain any existential quantifier. We proceed by structural induction along the following cases:

C1. $\phi(x)$ is an atomic formula. Then ϕ it is either a unary numerical predicate or a letter predicate; in both cases, a small circuit of linear size can be produced, showing the claim.

C2. $\phi(x)$ is a Boolean combination of subformulas. We can construct a circuit for $\phi(x)$ based on the circuits for the subformulas. For instance, with $\phi(x) \equiv \psi(x) \vee \theta(x)$ and circuits C_ψ and C_θ for ψ and θ, the output $C^{(i)}$ is the $\vee$ of $C_\psi^{(i)}$ and $C_\theta^{(i)}$. The number of wires of C is then that of C_ψ plus that of C_θ, and the extra $2n$ wires for the new connections, showing the claim.

C3. $\phi(x) = (\forall y)[\psi(y)]$, so that ϕ actually has no free variable. The induction hypothesis provides a circuit C_ψ with n outputs. The output of C, which is valid for any value of x, simply is the conjunction of the outputs of C_ψ, which adds linearly many wires, showing the claim.

C4. $\phi(x) = (\forall y)[\psi(x, y)]$, this is the nontrivial case. Elementary symbolic manipulation on ϕ shows that it can be written as (see [9, Theorem 2] for details):

$$\bigwedge_{\ell=1,\dots,k} \gamma_\ell(x) \vee (\forall y)[\delta_\ell(x, y) \vee \theta_\ell(y)],$$

where, for every ℓ, $\delta_\ell(x,y)$ is a Boolean combination of numerical predicates, each of them using both x and y. Additionally, all the $\gamma_\ell, \delta_\ell, \theta_\ell$ are structurally simpler than ϕ, so that we can apply the induction hypothesis on these formulas.
By induction hypothesis and Case 2 (Boolean combinations), we simply have to show that a formula of the form

$$\phi(x) = (\forall y)[\delta(x,y) \vee \theta(y)]$$

admits a WLAC^0 circuit, where δ is, as above, a Boolean combination of numerical predicates, all using both x and y. Note that the latter implies that δ is a Boolean combination of $x = y+k$, with k constant, and $x < y$ and the same with x and y swapped. The induction hypothesis provides a circuit C_θ for θ, but we cannot simply construct a circuit for δ, since it would have n^2 outputs, one for each pair of positions. However, for each $i \in \{1,\dots,n\}$, define

$$Y_i = \{j \in \{1,\dots,n\} \mid a^n \models \neg\delta(i,j)\},$$

for some $a \in A$ (since δ does not have any letter predicate, the choice of letter is not important). By construction, we have, for any word $w \in A^n$ and $i \in \{1,\dots,n\}$:

$$w \models \phi(i) \text{ iff } w \models \bigwedge_{j \in Y_i} \theta(j). \tag{1}$$

Note that when constructing a circuit C for ϕ, we cannot wire $C^{(i)}$ as the logical-and of the outputs $C_\theta^{(j)}$ with $j \in Y_i$. Indeed, although this would be semantically correct, this output gate would require a linear number of wires and we need n such outputs. We thus investigate the set Y_i more precisely.
Because δ only contains the $+1$ and $<$ predicates, each set Y_i is the union of a finite set whose size does not depend on n, a (possibly empty) starting segment of $\{1,\dots,n\}$, and a (possibly empty) finishing segment of $\{1,\dots,n\}$; in symbols, for any i there is a set F_i and two values s_i, t_i such that:

$$Y_i = F_i \cup \{j \mid j \le s_i \vee t_i \le j\} \cap \{1,\dots,n\}. \tag{2}$$

(We allow these values to lie outside of $\{1,\dots,n\}$ to make the corresponding part empty.)
We rely on the existence of the following WLAC^0 circuits:

▷ Claim 7 (*P*refix-and and *S*uffix-and circuits [6]). For any n, there is a circuit P with n inputs, n outputs, and a linear number of wires, such that $P^{(i)}$ computes the logical-and of the i *first* inputs. Similarly, there is a circuit S that does the same but for the i *last* inputs.

We can now start the construction of a circuit C for ϕ. Recall that C_θ is the circuit for θ and let us plug its n outputs, corresponding to different values of y, as inputs of both P and S. This contributes $2n$ new wires, in addition to those of P and S. Combining Equations 1 and 2, we obtain that the i-th output of C should compute:

$$\begin{aligned}\bigwedge_{j \in Y_i} C_\theta^{(j)} &\equiv \left(\bigwedge_{j \in F_i} C_\theta^{(j)}\right) \wedge \left(\bigwedge_{j \le s_i} C_\theta^{(j)}\right) \wedge \left(\bigwedge_{j \ge t_i} C_\theta^{(j)}\right) \\ &\equiv \left(\bigwedge_{j \in F_i} C_\theta^{(j)}\right) \wedge P^{(s_i)} \wedge S^{(t_i)}.\end{aligned}$$

Since the size of F_i only depends on the formula, the last expression can be wired using a constant number of wires. This concludes the construction.

◀

3.3 Back to algebra

In order to finish our chain of inclusions, we will study the class $\mathcal{QLDA}$ on its own in this section and show that it can be characterized by the languages that are *not* in it. For the rest of this paper, we let $K = (c + ac^*b)^*$, a regular language with a neutral letter that was shown to lie outside of WLAC^0 by Koucký, Pudlák, and Thérien [10].

▶ **Lemma 8.** *Let $\mathcal{V}$ be an lm-variety of languages. If $\mathcal{QLDA} \subseteq \mathcal{V} \subseteq \mathcal{QA}$ and $K \notin \mathcal{V}$, then $\mathcal{V} = \mathcal{QLDA}$.*

Proof. Let $\mathbf{V}$ such that $\mathbf{QLDA} \subseteq \mathbf{V} \subseteq \mathbf{QA}$. We will show that if $\mathbf{QLDA} \subsetneq \mathbf{V}$ then K' is in $\mathcal{V}$ where:

$$K' = (c + ac^*b)^*ac^* = (K \cap \{a, b, c\}^*b)b^{-1}.$$

Naturally, K' is in $\mathcal{V}$ if and only if K is; we use K' to slightly simplify our presentation.

Let $\mu\colon A^* \to M$ be in $\mathbf{V}$ but not in $\mathbf{QLDA}$, we will derive from μ a stamp in $\mathbf{V}$ that recognizes K'. Let $\rho\colon (A^s)^* \to S(\mu)$ be its stable stamp, which is in $\mathbf{A}$ since $\mu \in \mathbf{QA}$.

Since $\rho \notin \mathbf{LDA}$, there is an $e \in \rho((A^s)^+)$ such that the local stamp ρ_e of ρ induced by e is not in $\mathbf{DA}$. In turn, this means there are two words $x, y \in (A^s)^*$ such that:

$$\rho_e((xy)^\omega x(xy)^\omega) \neq \rho_e((xy)^\omega),$$

for succinctness, we will identify in the rest of this proof x with $\rho_e(x)$, and similarly for y.

Note that $(xy)^\omega x \in S(\mu)$, and since $S(\mu) = \mu(A^s)$, this means there is a word $u \in A^s$ such that $\mu(u) = (xy)^\omega x$. Similarly, there are words $v, w \in A^s$ such that $\mu(v) = y(xy)^\omega$ and $\mu(w) = e$.

Let $\phi\colon \{a, b, c\}^* \to A^*$ be an lm-morphism defined by $\phi(a) = u, \phi(b) = v, \phi(c) = w$ and finally combine it with μ to form $\eta\colon A^* \to eS(\mu)e = \mu \circ \phi$. We claim that η recognizes K'. Note that $\eta \in \mathbf{V} \cap \mathbf{A}$.

We first derive some inequalities and a few identities about η; inequalities characterize what η can distinguish, while identities exemplify what it cannot distinguish. Our inequalities mostly say that η can distinguish one letter from two.

- η can distinguish the words aab and ab, that is, $\eta(aab) \neq \eta(ab)$, indeed:

$$\begin{aligned}
\eta(aab) &= (xy)^\omega x \cdot (xy)^\omega x \cdot y(xy)^\omega \\
&= (xy)^\omega x \cdot (xy)^{(2\omega+1)} \\
&= (xy)^\omega x(xy)^\omega. && \text{(Since } \rho \in \mathbf{A}\text{)} \\
\eta(ab) &= (xy)^\omega x \cdot y(xy)^\omega \\
&= (xy)^\omega, && \text{(Similarly, since } \rho \in \mathbf{A}\text{)}
\end{aligned}$$

  and we have that $(xy)^\omega x(xy)^\omega \neq (xy)^\omega$.
- The previous point immediately implies that $\eta(a) \neq \eta(aa)$ and $\eta(b) \neq \eta(ab)$. It also implies that $\eta(a) \neq \eta(ab)$, since otherwise, as $\eta(abab) = \eta(ab)$, we would get that $\eta(ab) = \eta(aab)$. These are the inequalities that we will use.

- η cannot distinguish a and aba, that is, $\eta(a) = \eta(aba)$:

$$\eta(aba) = (xy)^\omega x \cdot y(xy)^\omega \cdot (xy)^\omega x = (xy)^\omega x = \eta(a).$$

  Note that this implies that $\eta(a) \neq \eta(b)$, since otherwise $\eta(a) = \eta(a^\omega)$ and thus $\eta(a) = \eta(aa) = \eta(ab)$, contradicting $\eta(a) \neq \eta(ab)$.
- η cannot distinguish b and bab, that is, $\eta(b) = \eta(bab)$:

$$\eta(bab) = y(xy)^\omega \cdot (xy)^\omega x \cdot y(xy)^\omega = y(xy)^\omega = \eta(b).$$

Let us now establish how η recognizes words in K'. Since c is a neutral letter, in the sense that it does not affect the value of η, we have for every word $w \in K'$ that $\eta(w) = \eta((ab)^k a)$ for some k. As $\eta(a) = \eta(aba)$, it holds that $\eta(w) = \eta(a)$. We claim that $K' = \eta^{-1}(\{\eta(a)\})$, with the inclusion from left to right holding by construction.

Assume for a contradiction that there is a word $w \notin K'$ such that $\eta(w) = \eta(a)$. By removing every c from w and repeatedly substituting b for bab and a for aba, we obtain a word w' such that $\eta(w') = \eta(a)$ and w' falls in one of the following cases, which all lead to a contradiction:

C1. $w' = a$. This is impossible since w would have been in K'.

C2. $w' = b$. This is impossible since $\eta(a) \neq \eta(b)$.

C3. $w' = r \cdot aa \cdot t$. It holds that:

$$\eta(a) = \eta(raat). \tag{3}$$

By iterating Equation 3 onto itself, we get $\eta(a) = \eta(r^\omega a(at)^\omega)$. By aperiodicity, we thus get that $\eta(a) = \eta(ra)$; we can substitute this into Equation 3 and obtain $\eta(a) = \eta(aat)$. Now we could have iterated Equation 3 onto itself in a different way and obtain $\eta(a) = \eta((ra)^\omega at^\omega)$, from which we deduce similarly that $\eta(a) = \eta(at)$. Substituting this further into Equation 3, we obtain that $\eta(a) = \eta(aa)$, a contradiction.

C4. $w' = r \cdot bb \cdot t$. We distinguish two cases:

C4.1. Assume $(xy)^\omega y(xy)^\omega = (xy)^\omega$. Consequently:

$$\eta(bb) = y(xy)^\omega \cdot y(xy)^w = y(xy)^\omega = \eta(b). \tag{4}$$

In this case, we can keep on reducing w' by replacing each bb with b, and apply one of C2 or C3 to reach a contradiction.

C4.2. Assume $(xy)^\omega y(xy)^\omega \neq (xy)^\omega$. From this, it holds that η can distinguish ab and abb:

$$\eta(abb) = (xy)^\omega x \cdot y(xy)^\omega \cdot y(xy)^\omega = (xy)^\omega y(xy)^\omega \neq (xy)^\omega = \eta(ab).$$

This immediately implies that $\eta(b) \neq \eta(bb)$; also, $\eta(a) \neq \eta(ba)$, since otherwise, $\eta(aab) = \eta(abab) = \eta(ab)$.

We assume we are not in Case 3, so that no aa appears in w'. As w' is fully reduced, so that neither aba nor bab appears in w', we are left with only a few subcases:

- $w' = bbb^k$ for some $k \geq 0$. Then $\eta(a) = \eta(bbb^k)$, hence $\eta(bab) = \eta(bbbb^k b)$. The left-hand side is $\eta(b)$, which is thus equal, by iterating, to $\eta(b^\omega)$. Hence, by aperiodicity, $\eta(b) = \eta(bb)$, a contradiction.
- $w' = abbb^k$ for some $k \geq 0$. Then $\eta(a) = \eta(abbb^k)$, and substituting, $\eta(a) = \eta(ab^\omega)$, showing, by aperiodicity, $\eta(a) = \eta(ab)$, a contradiction.
- $w' = b^k bba$ for some $k \geq 0$. Then $\eta(a) = \eta(b^k bba)$, and substituting, $\eta(a) = \eta(b^\omega a)$, showing, by aperiodicity, $\eta(a) = \eta(ba)$, a contradiction.

- $w' = ab^kbba$ for some $k \geq 0$. Then $\eta(a) = \eta(ab^kbba)$, hence $\eta(ab) = \eta(ab^kbbab) = \eta(ab^kbb)$. Iterating, this results in $\eta(ab) = \eta(ab^\omega)$, hence, by aperiodicity, $\eta(ab) = \eta(abb)$, a contradiction.

◀

We can make this statement more salient by showing a characterization of lm-varieties included in $\mathcal{QA}$ by exclusion and obtain:

▶ **Theorem 9.** *$\mathcal{QLDA}$ is the largest lm-variety of languages that does not contain K nor the languages, for any $d \geq 2$, $L_d = \{w \in \{a,c\}^* \mid w \text{ has } 0 \mod d \text{ letters } a\}$.*

Proof. We show that $\mathcal{QA}$ is the largest lm-variety that excludes the languages L_d. This will show that any variety that excludes these languages is in $\mathcal{QA}$, and in turn that if it excludes K, by Lemma 5, it is in fact $\mathcal{QLDA}$.

Let $\mathcal{V}$ be an lm-variety that excludes the languages L_d and assume there is a stamp $\mu\colon A^* \to M$ in $\mathbf{V} \setminus \mathbf{QA}$. Let ρ be its stable stamp which, by assumption, is not in $\mathbf{A}$. This means that there is a word $x \in A^s$ such that $\rho(x^\omega) \neq \rho(x^{\omega+1})$. Let $d \geq 2$ be the smallest value such that $\rho(x^\omega) = \rho(x^{\omega+d})$.

Since the image of ρ is $\mu(A^s)$, we can find two words $u, v \in A^s$ such that $\rho(u) = \rho(x^\omega)$ and $\rho(v) = \rho(x^{\omega+1})$. Let $\phi\colon \{a,c\}^* \to A^*$ be the lm-morphism defined by $\phi(c) = u$ and $\phi(a) = v$, and write $\eta = \mu \circ \phi = \rho \circ \phi$. Since $\mathbf{V}$ is an lm-variety, $\eta \in \mathbf{V}$.

Note that $\eta(ca) = \eta(ac) = \rho(x^\omega x^{\omega+1}) = \rho(x^{2\omega}x) = \rho(x^{\omega+1}) = \eta(a)$, hence c is a neutral letter for η, in the sense that adding or removing it from a word does not change the value of η. Moreover:

$$\begin{aligned}\eta(a^k) &= \rho((x^{\omega+1})^k)\\ &= \rho(x^{k\omega+k})\\ &= \rho(x^{\omega+k}).\end{aligned}$$

In particular, $\eta(a^k) = \eta(c)$ iff $k \equiv 0 \mod d$. Hence $\eta^{-1}(\{\eta(c)\})$ is L_d, a contradiction. ◀

3.4 Closing the circle: from circuits to algebra

So far, we have established the following chain of inclusions:

$$\mathcal{QLDA} \subseteq \mathrm{FO}^2[\mathrm{reg}] \subseteq \mathrm{WLAC}^0 \cap \mathrm{Reg},$$

and in addition, since $\mathrm{AC}^0 \cap \mathrm{Reg} = \mathcal{QA}$, we have that $\mathrm{WLAC}^0 \cap \mathrm{Reg} \subseteq \mathcal{QA}$. Hence if we could use Lemma 5 on $\mathrm{WLAC}^0 \cap \mathrm{Reg}$, we would be done. Since $K \notin \mathrm{WLAC}^0$, the only missing piece is:

▶ **Lemma 10.** $\mathrm{WLAC}^0 \cap \mathrm{Reg}$ *is an lm-variety.*

Proof. The proof is standard; in fact, WLAC^0 itself is an lm-variety. We need to show that WLAC^0 is closed under the Boolean operations (which is immediate), quotient, and inverse lm-morphism. Let $L \in \mathrm{WLAC}^0$ and (C_n) be its circuit family.

- Closure under quotient. Let a be a letter, we construct a circuit family for $a^{-1}L$, the symmetric case being similar. To do so, we simply import, for input length n, the circuit C_{n+1} and hardwire its first input to a, while its other inputs are directly connected to the inputs of the global circuit. The number of wires is that of C_{n+1} plus $n+1$ wires, which is still linear.

- Closure under inverse lm-morphism. Let $\phi\colon A^* \to B^*$ be an lm-morphism and k be such that $\phi(A) \subseteq B^k$. We construct a circuit for $\phi^{-1}(L)$ by reading the inputs over A. Each input is converted into k inputs over B and the resulting word is plugged into the circuit C_{kn}. Since k is a constant, we are only adding a linear number of wires to C_{kn}, which contains itself only a linear number of wires.

◀

▶ **Theorem 11.** $\mathcal{QLDA} = \mathrm{FO}^2[\mathrm{reg}] = \mathrm{WLAC}^0 \cap \mathrm{Reg}$.

4 Applications

4.1 Decidability

Given a regular language, can we decide whether it belongs to WLAC^0? Questions of this nature come with an important application: if we can decide them, it is often the case that we can also *produce* a circuit in the class under study. For instance, this is the case for AC^0 — note that all regular languages belong to the class NC^1 of bounded fan-in, log-depth, poly-size circuits and $\mathrm{AC}^0 \subsetneq \mathrm{NC}^1$, thus a AC^0 circuit would be more efficient than the "default" NC^1 circuit.

▶ **Theorem 12.** *It is decidable, given a regular language, whether it belongs to* WLAC^0. *Moreover, a* WLAC^0 *circuit can be built for the language.*

Proof. Given a regular language, we can compute its so-called *syntactic stamp* μ, which will belong to **QLDA** iff the language belongs to WLAC^0 by Theorem 8. Deciding this, in turn, requires to compute the stable stamp of μ and then its local stamps, then checking that they belong to **DA**. The fact that **DA** is decidable is a classical result [18].For the "moreover" part, we can enumerate all $\mathrm{FO}_2[\mathrm{reg}]$ formulas and check whether their (regular) language corresponds to the input language. The transformation of a $\mathrm{FO}^2[\mathrm{reg}]$ formula into a WLAC^0 circuit is then given in the proof of Lemma 3. ◀

We note that the characterization of $\mathrm{WLAC}^0 \cap \mathrm{Reg} \cap \mathcal{NL}$ (Theorem 1) is also effective, but one cannot reduce membership in $\mathrm{WLAC}^0 \cap \mathrm{Reg}$ to membership in $\mathrm{WLAC}^0 \cap \mathrm{Reg} \cap \mathcal{NL}$ just by adding a neutral letter. Indeed, the language $(ab)^*$ is in $\mathrm{WLAC}^0 \cap \mathrm{Reg}$, but adding a neutral letter to it results in K.

4.2 $\mathrm{LAC}^0 \cap \mathrm{Reg}$, Straubing and Crane Beach properties

One of the most tantalizing open question in the study of the linear restrictions of AC^0 is whether $\mathrm{LAC}^0 \cap \mathrm{Reg} = \mathrm{WLAC}^0 \cap \mathrm{Reg}$ (Section 1 in the Introduction) — this is widely believed to be true. We note that this hinges on the language K:

▶ **Proposition 13.** *If* $K \notin \mathrm{LAC}^0$, *then* $\mathrm{LAC}^0 \cap \mathrm{Reg} = \mathrm{WLAC}^0 \cap \mathrm{Reg}$.

Proof. It is not hard to show, along the lines of Lemma 7, that $\mathrm{LAC}^0 \cap \mathrm{Reg}$ is an lm-variety — in fact, LAC^0 itself is. Hence if $K \notin \mathrm{LAC}^0$, and since $\mathrm{LAC}^0 \cap \mathrm{Reg} \subseteq \mathcal{QA}$, Lemma 5 implies that $\mathrm{LAC}^0 \cap \mathrm{Reg} = \mathcal{QLDA}$, which is equal to $\mathrm{WLAC}^0 \cap \mathrm{Reg}$. ◀

Similarly, showing $K \notin \mathrm{FO}^2[\mathrm{arb}]$ would elucidate the Straubing property of FO^2 (Section 1 in the Introduction). Indeed, since $\mathrm{LAC}^0 = \mathrm{FO}^2[\mathrm{arb}]$, the previous proposition can be written as:

▶ **Proposition 14.** *If* $K \notin \mathrm{FO}^2[\mathrm{arb}]$*, then* $\mathrm{FO}^2[\mathrm{arb}]$ *has the Straubing property:*

$$\mathrm{FO}^2[\mathrm{arb}] \cap \mathrm{Reg} = \mathrm{FO}^2[\mathrm{reg}].$$

Finally, we come short of showing the Crane Beach property for $\mathrm{FO}^2[\mathrm{arb}]$ (Section 1) but we show the property holds for $\mathrm{FO}^2[\mathrm{reg}]$:

▶ **Proposition 15.** $\mathrm{FO}^2[\mathrm{reg}]$ *has the Crane Beach property:*

$$\mathrm{FO}^2[\mathrm{reg}] \cap \mathcal{NL} \subseteq \mathrm{FO}^2[<].$$

Proof. This is a direct consequence of Theorem 1 and Theorem 8. ◀

4.3 Bounded-depth Dyck languages

The language $(ab)^*$ can be seen as the Dyck language of depth 1: we can interpret a as opening parenthesis and b as closing. More generally, we can define, for any $k > 1$:

$$D_1^{(1)} = (ab)^*,$$
$$D_1^{(k)} = (a \cdot D_1^{(k-1)} \cdot b)^*.$$

These are bounded-depth Dyck languages, in the sense that the classical Dyck language D_1 is the union of them all.

These languages play an interesting role in the study of FO and AC^0; indeed, one can show that $D_1^{(k)}$ separates $\mathrm{FO}[<]$ with quantifier-depth k and $k+1$. This also holds if we were to add a neutral letter to these languages [5, 15]. It is open whether these languages separate the depth hierarchy of AC^0.

As mentioned before, if we were to add a neutral letter to $D_1^{(1)} = (ab)^*$, we would obtain the language K and readily be out of WLAC^0. However, $D_1^{(1)}$ itself is in WLAC^0. In fact:

▶ **Proposition 16.** $D_1^{(k)}$ *is in* WLAC^0 *if and only if* k *is* 1 *or* 2*.*

Proof. The case $k = 1$ is an easy exercise. For the case $k = 2$, we note that there are two types of letters a: those at depth 1 ($\underline{a}abb$) and those at depth 2 ($a\underline{a}bb$). Importantly, those at depth 1 can only appear in odd positions (starting with 1), while those at depth 2 can only appear in even positions. The symmetric consideration holds for bs. Hence a nonempty word is in $D_1^{(2)}$ iff all of the following hold:

- It starts with a and ends with b,
- Each a in even position (depth 2) is followed by a b,
- Each b in even position (depth 1) but the last is followed by an a,
- The word is of even length.

Using the successor and modulo predicates, these three properties can easily be formulated in $\mathrm{FO}^2[\mathrm{reg}]$, showing that $D_1^{(2)} \in \mathrm{WLAC}^0$.

For the case $k = 3$, we show $D_1^{(3)} \notin \mathrm{WLAC}^0$. As $\mathrm{WLAC}^0 \cap \mathrm{Reg}$ is decidable (Theorem 9), we could implement a procedure that checks this. This is done in the online tool *Semigroup* by the second author, available at `https://paperman.name/semigroup`, which indicates that $D_1^{(3)} \notin \mathcal{QLDA}$.

For completeness, we present an elementary ad-hoc proof. Let $\phi\colon \{a,b,c\}^* \to \{a,b\}^*$ be the lm-morphism defined by $\phi(a) = aa, \phi(b) = bb, \phi(c) = ab$. Define $L = \phi^{-1}(D_1^{(3)})$. Clearly, no word in L can contain aa nor bb. Additionally, c is neutral in L, and every nonempty word in L that does not contain c starts with a and ends with b. This shows that L is in fact

$K = (c + ac^*b)^*$. In turn, since WLAC^0 is an lm-variety, if $D_1^{(3)}$ is in WLAC^0, so is K; this implies that $D_1^{(3)} \notin \mathrm{WLAC}^0$.

For the case $k > 3$, assume that $D_1^{(k)}$ is in WLAC^0 for a contradiction. Since WLAC^0 is an lm-variety, it is closed under quotient, implying that $a^{-1}D_1^{(k)}b^{-1}$ is in WLAC^0. This latter language is actually $D_1^{(k-1)}$. Iterating the quotients, this shows that if $D_1^{(k)}$ is in WLAC^0 for $k > 3$, then $D_1^{(3)}$ is in WLAC^0, a contradiction with the case $k = 3$ above. ◀

We note that if the languages $D_1^{(k)}$ were equipped with depth information, then they would all be in WLAC^0, showing that the difficulty with these languages is not the nesting, but the fact that a letter can appear at different depths. This is in contrast with the case where a neutral letter is added, in which no depth information can be provided to keep the language in WLAC^0. Formally, define, for all $k > 1$:

$$R_1^{(1)} = (a_1 b_1)^*,$$
$$R_1^{(k)} = (a_k \cdot R_1^{(k-1)} \cdot b_k)^*.$$

The language $R_1^{(k)}$ is over the alphabet $\{a_1, b_1, \ldots, a_k, b_k\}$. It holds that:

▶ **Proposition 17.** *For every k, $R_1^{(k)}$ is in WLAC^0.*

5 Conclusion

Our main contribution is an exact algebraic and logical characterization of the regular languages of wire linear AC^0. We noted that Open Problems 1 and 1 hinge on the membership of a single language in LAC^0: $K = (c + ac^*b)^*$.

We deduced from the characterization a decidability result that follows a long line of tradition: one can decide if a regular languages belong to WLAC^0. An often overlooked application of this type of decidability results is that they can be used to provide efficient electronic circuits automatically for regular expressions. It is conceivable that compilation procedures can be devised, targeting for instance FPGAs (field programmable gate arrays) or other computing hardware exhibiting parallelism.

Acknowledgments. We thank Luc Dartois who contributed to some of the proofs in this paper and Corentin Barlois for expert proofreading. We naturally thank Klaus-Jörn for a fantastic few years in Tübingen and wish him a happy 70th birthday.

References

1 Jorge Almeida. A syntactical proof of locality of DA. *International Journal of Algebra and Computation*, 06(02):165–177, 1996. doi:10.1142/S021819679600009X.

2 D. A. Mix Barrington, K. Compton, H. Straubing, and D. Thérien. Regular languages in NC^1. *J. Computer and System Sciences*, 44:478–499, 1992.

3 Christoph Behle, Andreas Krebs, Klaus-Jörn Lange, and Pierre McKenzie. The lower reaches of circuit uniformity. In Branislav Rovan, Vladimiro Sassone, and Peter Widmayer, editors, *Mathematical Foundations of Computer Science 2012*, pages 590–602, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

4 Christoph Behle and Klaus-Jörn Lange. Fo[<]-uniformity. In *21st Annual IEEE Conference on Computational Complexity (CCC 2006), 16-20 July 2006, Prague, Czech Republic*, pages 183–189. IEEE Computer Society, 2006. doi:10.1109/CCC.2006.20.

5 J.A. Brzozowski and R. Knast. The dot-depth hierarchy of star-free languages is infinite. *Journal of Computer and System Sciences*, 16(1):37–55, 1978. URL: `https://www.sciencedirect.com/science/article/pii/0022000078900491`, `doi:https://doi.org/10.1016/0022-0000(78)90049-1`.

6 Ashok K. Chandra, Steven Fortune, and Richard Lipton. Unbounded fan-in circuits and associative functions. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, STOC '83, page 52–60, New York, NY, USA, 1983. Association for Computing Machinery. `doi:10.1145/800061.808732`.

7 Luc Dartois and Charles Paperman. Alternation hierarchies of first order logic with regular predicates. In Adrian Kosowski and Igor Walukiewicz, editors, *Fundamentals of Computation Theory*, pages 160–172, Cham, 2015. Springer International Publishing.

8 Michael Hahn, Andreas Krebs, Klaus-Jörn Lange, and Michael Ludwig. Visibly counter languages and the structure of NC^1. In Giuseppe F. Italiano, Giovanni Pighizzini, and Donald T. Sannella, editors, *Mathematical Foundations of Computer Science 2015*, pages 384–394, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

9 M. Koucky, S. Poloczek, C. Lautemann, and D. Therien. Circuit lower bounds via ehrenfeucht-fraisse games. In *21st Annual IEEE Conference on Computational Complexity (CCC'06)*, pages 12 pp.–201, 2006. `doi:10.1109/CCC.2006.12`.

10 Michal Koucký, Pavel Pudlák, and Denis Thérien. Bounded-depth circuits: Separating wires from gates. In *STOC*, pages 257–265. ACM, 2005. `doi:10.1145/1060590.1060629`.

11 Andreas Krebs, Klaus-Jörn Lange, and Michael Ludwig. Visibly counter languages and constant depth circuits. In Ernst W. Mayr and Nicolas Ollinger, editors, *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, volume 30 of *LIPIcs*, pages 594–607. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. `doi:10.4230/LIPIcs.STACS.2015.594`.

12 Andreas Krebs, Klaus-Jörn Lange, and Stephanie Reifferscheid. Characterizing tc^0 in terms of infinite groups. In Volker Diekert and Bruno Durand, editors, *STACS 2005, 22nd Annual Symposium on Theoretical Aspects of Computer Science, Stuttgart, Germany, February 24-26, 2005, Proceedings*, volume 3404 of *Lecture Notes in Computer Science*, pages 496–507. Springer, 2005. `doi:10.1007/978-3-540-31856-9\_41`.

13 Manfred Kufleitner and Alexander Lauser. Quantifier Alternation in Two-Variable First-Order Logic with Successor Is Decidable. In Natacha Portier and Thomas Wilke, editors, *30th International Symposium on Theoretical Aspects of Computer Science (STACS 2013)*, volume 20 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 305–316, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.STACS.2013.305`.

14 Klaus-Jörn Lange. Some results on majority quantifiers over words. In *19th Annual IEEE Conference on Computational Complexity (CCC 2004), 21-24 June 2004, Amherst, MA, USA*, pages 123–129. IEEE Computer Society, 2004. `doi:10.1109/CCC.2004.1313817`.

15 Charles Paperman. *Circuits booléens, prédicats modulaires et langages réguliers*. PhD thesis, Université Paris Diderot, 2013.

16 Jean-Éric Pin and Howard Straubing. Some results on C-varieties. *RAIRO-Theoretical Informatics and Applications*, 39(01):239–262, 2005.

17 H. Straubing. *Finite Automata, Formal Logic, and Circuit Complexity*. Birkhäuser, Boston, 1994.

18 Pascal Tesson and Denis Thérien. Diamonds are forever: The variety DA. In *Semigroups, Algorithms, Automata and Languages*, pages 475–500. World Scientific, 2002.

19 D. Thérien and T. Wilke. Over words, two variables are as powerful as one quantifier alternation: $FO^2 = \Sigma_2 \cap \Pi_2$. In *Proceedings 30th Symposium on Theory of Computing*, pages 234–240, New York, 1998. ACM Press.