The Descriptive Complexity Approach to LOGCFL

Clemens Lautemann,* Pierre McKenzie,† Thomas Schwentick,* Heribert Vollmer‡

Abstract

Building upon the known generalized-quantifier-based first-order characterization of LOGCFL, we lay the groundwork for a deeper investigation. Specifically, we examine subclasses of LOGCFL arising from varying the arity and nesting of *groupoidal* quantifiers in first-order logic with linear order. Our work extends the elaborate theory relating *monoidal* quantifiers to NC^1 and its subclasses. In the absence of arithmetical predicates for plus and times (equivalently: in the absence of the BIT predicate), we resolve the main issues: we show in particular that no single outermost unary groupoidal quantifier with FO can capture all the context-free languages, and we obtain the surprising result that a variant of Greibach's "hardest context-free language" is LOGCFL-complete under quantifier-free reductions without arithmetic. We then prove that FO with unary groupoidal quantifiers is strictly more expressive with predicates for plus and times than without. Considering a particular groupoidal quantifier, we prove that first-order logic with majority of pairs is strictly more expressive than first-order with majority of individuals. As a technical tool of independent interest, we define the notion of an aperiodic nondeterministic finite automaton and prove that FO translations are precisely the mappings computed by single-valued aperiodic nondeterministic finite transducers.

Keywords: finite model theory, descriptive complexity, computational complexity, automata and formal languages

1 Introduction

In *Finite Automata, Formal Logic, and Circuit Complexity* [Str94], Howard Straubing surveys an elegant theory relating finite semigroup theory, first-order logic, and computational complexity. The gist of this theory is that questions about the structure of the complexity class NC^1, defined via logarithmic-depth bounded fan-in Boolean circuits, can be translated back and forth into questions about the expressibility of first-order logic augmented with new predicates and quantifiers. Such a

*Institut für Informatik, Johannes-Gutenberg-Universität Mainz, 55099 Mainz, Germany.

†Informatique et recherche opérationnelle, Université de Montréal, C.P. 6128, Succ. Centre-Ville, Montréal (Québec), H3C 3J7 Canada. Research performed while on leave at the Universität Tübingen. Supported by the (German) DFG, the (Canadian) NSERC and the (Québec) FCAR.

‡Theoretische Informatik, Universität Würzburg, Am Hubland, 97074 Würzburg, Germany.

translation provides new insights, makes tools from one field available in the other, suggests tractable refinements to the hard open questions in the separate fields, and puts the obstacles to further progress in a clear perspective.

The unresolved strict containment in NC^1 of the class ACC^0, defined from bounded-depth polynomial-size unbounded fan-in circuits over {AND, OR, MOD}, remains a barrier since the work of Smolensky [Smo87]. Nevertheless, significant progress was made in

– understanding the power of the arithmetical predicates (plus and times) and the related circuit uniformity issues [BIS90],

– describing the regular languages within subclasses of NC^1 [BCST92, PMT91], and

– identifying the all-important role of the interplay between arbitrary and regular numerical predicates in the status of the ACC^0 versus NC^1 question [Str94, p. 169, Conjecture IX.3.4].

Starting with the work of Barrington and Thérien [Bar89, BT88], word problems for algebraic structures have turned out to correspond to complexity classes in the range between AC^0 and NC^1. The word problem for a group G is essentially defined as the problem of computing the product of a sequence of elements of G. Barrington has shown that the word problem for any nonsolvable group is complete for NC^1 while Barrington and Thérien have related the word problem for solvable monoids to the class ACC^0, and the word problem for aperiodic monoids to the class AC^0. Barrington, Immerman and Straubing [BIS90] then introduced the notion of a *monoidal* quantifier; these authors noted that the hardness of a word problem for a monoid M can equivalently be stated as an expressibility result using first-order logic augmented with a monoidal quantifier for M. Loosely speaking, such a quantifier is a constrained "oracle call" to the word problem for M.

Bédard, Lemieux and McKenzie [BLM93] later turned to more general algebraic structures, namely *groupoids*. A groupoid G is a set with a binary operation on which no constraint—such as associativity or commutativity—is placed. The word problem for G is the set of all those sequences of elements of G that can be bracketed to evaluate to a fixed element of G. It is not hard to see that any context-free language is the word problem of some groupoid, and that any groupoid word problem is context-free (see [BLM93, Lemma 3.1]).

Bédard et al. showed that there is a fixed finite groupoid whose word problem is complete for the class LOGCFL of languages reducible in logarithmic space to a context-free language [Coo71, Sud78]. It followed that LOGCFL, a well-studied class which contains nondeterministic logarithmic space [Sud78] and is presumably much larger than NC^1, can be described by first-order logic augmented with *groupoidal* quantifiers. These quantifiers can be defined formally as Lindström quantifiers [Lin66] for context-free languages.

In this article, we take up the groupoidal first-order characterization of LOGCFL, and initiate an investigation of classes in the range between NC^1 and LOGCFL from the viewpoint of descriptive complexity. The rationale for this study, which encompasses the study of subclasses of NC^1, is that tools from logic might be of use in ultimately elucidating the structure of LOGCFL. We do not claim new separations of the major subclasses of LOGCFL here. But we make a first step, in effect settling necessary preliminary questions afforded by the first-order framework.

Our precise results concern the relative expressiveness of first-order formulas with ordering (written FO), interpreted over finite strings, and with:

(1) nested versus unnested groupoidal quantifiers,

(2) unary versus non-unary groupoidal quantifiers,

(3) the presence versus the absence of the numerical predicates plus and times.

Feature (3) was the focus of an important part of the work by Barrington, Immerman and Straubing [BIS90] on uniformity within NC^1. Feature (2) was also considered, to a lesser extent, by the same authors, who left open the question of whether the "majority-of-pairs" quantifier could be simulated by a unary majority quantifier in the absence of the BIT predicate (equivalently: the absence of arithmetical predicates for plus and times) [BIS90, p. 297]. Feature (1) is akin to comparing many-one reducibility with Turing reducibility in traditional complexity theory.

Here we examine all combinations of features (1), (2) and (3). Our separation results are summarized on Fig. 1 in Sect. 5. In the *absence of arithmetical predicates*, we are able to determine the following relationships:

(I) FO to which a single unary groupoidal quantifier is applied, written $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}$, captures the CFLs, and is strictly less expressive than FO with nested unary quantifiers, written $\mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}})$, which in its turn is strictly weaker than LOGCFL. A consequence of this result, as we will see, is an answer to the above mentioned open question from [BIS90]: We show that first-order logic with the majority-of-pairs quantifier is strictly more expressive than first-order logic with majority of individuals.[1]

(II) No single groupoid G captures all the CFLs as $Q^{\mathrm{un}}_G\mathrm{FO}$, i.e., as FO to which the single unary groupoidal quantifier Q^{un}_G is applied.

(III) FO to which a single *non-unary* groupoidal quantifier is applied, written $Q_{\mathrm{Grp}}\mathrm{FO}$, captures LOGCFL; our proof implies, remarkably, that adding a padding symbol to Greibach's hardest context-free language [Gre73], see also [Har78], yields a language that is LOGCFL-complete under quantifier-free projections without arithmetic.

Greibach's hardest context-free language H is a so called non-deterministic version of the Dyck-language D_2, the language of all syntactically correct sequences

[1]Independently from the present work, the question from [BIS90] was solved in a draft by S. Lindell [Lin95].

consisting of letters for two types of parentheses; for a formal definition, refer to, e.g., [Har78, p. 326] or [ABB97, p. 136]. Greibach showed that every context-free language can be obtained as the inverse image of H under an ϵ-free homomorphism. In other words: Every context-free language reduces to H under some homomorphism (thus the name "hardest context-free language").

LOGCFL is the closure of CFL under logarithmic-space bounded reductions. But our result (III) shows that all of LOGCFL is even reducible to one single context-free language under quantifier-free reductions without arithmetic – a notion much weaker than logarithmic-space reductions. In fact, these reductuions are even weaker than the AC^0 reductions studied in [AAI$^+$97], which, in turn, are known to be very restrictive, as witnessed by the fact that for many complexity classes (precisely, all classes closed under TC^0 reductions), the sets complete under AC^0 reductions are isomorphic (under depth-3 AC^0 isomorphisms) [AAI$^+$97].

When *arithmetical predicates for plus and times are present*, (extensions of) first-order logic can be quite expressive. In the setting of monoidal quantifiers [BIS90], FO with arithmetic is known to capture uniform circuit classes, notably uniform ACC^0, which have not yet been separated from NC^1. We face a similar situation here: of course, with non-unary groupoidal quantifiers we can still describe LOGCFL. However, since with arithmetic unary groupoidal quantifiers are enough to capture TC^0, a separation of $\mathrm{FO}(+,\times,Q^{\mathrm{un}}_{\mathrm{Grp}})$ from $\mathrm{FO}(+,\times,Q_{\mathrm{Grp}})$ would seem to require a major breakthrough. We are able to attest to the strength of the arithmetical predicates in the setting of unary quantifiers, proving that:

(IV) $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO} \subsetneq Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}(+,\times)$, i.e., (trivially) some non-context-free languages are expressible using arithmetic and a single unary groupoidal quantifier (clearly, also $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO} \not\supseteq \mathrm{FO}(+,\times)$ holds),

(V) $\mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}}) \subsetneq \mathrm{FO}(+,\times,Q^{\mathrm{un}}_{\mathrm{Grp}})$, i.e., (more interestingly) arithmetical predicates add expressivity even when unary groupoidal quantifiers can be nested.

We also develop a technical tool of independent interest, in the form of an aperiodic (a. k. a. group-free, a. k. a. counter-free) nondeterministic finite automaton. Aperiodicity has been studied intensively, most notably in connection with the star-free regular languages [Sch65], but, to the best of our knowledge, always in a deterministic context. The natural extension of aperiodicity to NFAs is such that an NFA A is aperiodic if and only if the DFA resulting from applying the subset construction to A is aperiodic. The usefulness of this notion lies in the fact, proved here, that first-order translations are precisely those mappings which are computable by single-valued aperiodic nondeterministic finite transducers.

Section 2 in this article describes our first-order framework and exhibits a link between standard formal language operations and unary generalized quantifiers. Section 3 introduces nondeterministic finite transducers and proves that they characterize first-order translations. Section 4 forms the bulk of the article and develops the relationships between our logic-based LOGCFL subclasses. Section 5 concludes with a number of suggestions on how to extend the results obtained here.

2 Preliminaries

2.1 Complexity theory

We assume familiarity with standard notions in formal languages, automata, and complexity theory, including classes such as L, NL, and P. REG and CFL refer to the regular and to the ϵ-free context-free languages respectively. The CFL results in this article could be adapted to treat the empty string ϵ in standard ways.

In this article, AC^0, ACC^0, TC^0, NC^1, and SAC^1 stand for the classes of languages recognized by *logtime uniform* families $(C_n)_{n\in\mathbb{N}}$ of polynomial-size circuits of the following kinds:

AC^0: the C_n may have NOT and unbounded fan-in AND/OR gates and constant depth;

ACC^0: the C_n additionally may have unbounded fan-in MOD_q gates, which are defined to output 1 if the number of their inputs that are 1 is a multiple of q (where q is fixed independent of n);

TC^0: the C_n may have NOT and unbounded fan-in MAJORITY gates, which are defined to output 1 if at least half their inputs are 1;

NC^1: the C_n may have NOT and bounded fan-in AND/OR gates and $O(\log n)$ depth;

SAC^1: the C_n may have NOT, bounded fan-in AND gates, unbounded fan-in OR gates, and $O(\log n)$ depth;

Logtime uniform means that the so called *direct connection language* of $(C_n)_{n\in\mathbb{N}}$, describing the structure of this family as a family of directed acyclic graphs, can be recognized by a deterministic Turing machine in time $O(\log n)$; for details about the subtleties involved in these issues, we refer the reader to, e.g., [Vol99]. In particular, for the class NC^1 another uniformity condition might be more natural, but this is of no concern to us here in this article, because our main results are about the class SAC^1. In fact, for the class SAC^1 there is a uniformity condition, logspace-uniformity, equivalent to the above, that can be described as follows: An SAC^1-family $(C_n)_{n\in\mathbb{N}}$ is *logspace-uniform*, if there is a deterministic Turing machine, operating in logarithmic space, that on an input of length n produces as output an encoding of C_n. This encoding of C_n is a sequence of the encodings of the gates of C_n. A gate is encoded by a tuple specifying its type and its predecessors.

In [Ven91] (c.f. also [Coo71] and[Sud78]) Venkateswaran showed that SAC^1 equals the class LOGCFL of languages that reduce in logarithmic space to some context-free language. Together with the separation of ACC^0 from AC^0 mentioned already [Ajt83, FSS84, Smo87] and the obvious relations among the classes defined, we thus obtain the following well-known inclusion chain:

$$\mathrm{AC}^0 \subsetneq \mathrm{ACC}^0 \subseteq \mathrm{TC}^0 \subseteq \mathrm{NC}^1 \subseteq \mathrm{NL} \subseteq \mathrm{LOGCFL} = \mathrm{SAC}^1 \subseteq \mathrm{P}.$$

2.2 The first-order framework

We consider first-order logic with linear order. We restrict our attention to *string signatures*, i.e., signatures of the form $\langle P_{a_1}, \ldots, P_{a_s} \rangle$, where all the predicates P_{a_i} are unary, and in every structure $\mathcal{A}$, $\mathcal{A} \models P_{a_i}(j)$ iff the jth symbol in the input is the letter a_i. Such structures are thus words over the alphabet $(a_1, \ldots, a_s)$, and first-order variables range over positions within such a word, i.e., from 1 to the word length n. The logic's linear order symbol refers to the numerical order on $\{1, \ldots, n\}$. For technical reasons to be motivated shortly, we also assume that every alphabet has a built-in linear order, and we write alphabets, as above, as sequences of symbols to indicate that order. Whenever $\Sigma = \{a_1, \ldots, a_s\}$ we call a formula of signature $\langle P_{a_1}, \ldots, P_{a_s} \rangle$ a *formula over* Σ.

Our basic formulas are built from variables in the usual way, using the Boolean connectives $\{\wedge, \vee, \neg\}$, the relevant predicates P_{a_i} together with $\{=, <\}$, the constants min and max, the quantifiers $\{\exists, \forall\}$, and parentheses.

FO is the set of all languages definable using first-order formulas as just described.

We use FO(+) to denote that additionally the predicate "$x + y = z$" (with the obvious semantics) is allowed. It is known that the numerical predicates definable in FO(+) (in the sense of [Str94]) are exactly the semi-linear sets (see [Har78, p. 231]).

A further extension of first-order logic allows the predicate "$x \times y = z$" (again with the obvious semantics). Our notation here is FO$(+, \times)$. In the literature, a binary predicate BIT(x, y), defined to be true iff the xth bit in the binary representation of y is 1, was also considered. It is known that, under our general assumption of the presence of a linear order, BIT and arithmetical predicates are equally expressive, in symbols: FO$(+, \times)$ = FO(BIT) (see [Imm99]). Hence, capturing a class of languages without arithmetic is the same as capturing the class without the BIT-predicate, an issue that has raised a lot of attention in the past [BIS90].

As a side remark, we note that it is even known that the BIT predicate alone is sufficient to express the linear order [DDLW98]. This implies that the identity FO$(+, \times)$ = FO(BIT) holds even when a linear order is not present in the logic.

Next, we define logically defined transformations on strings and the notion of Lindström quantifiers.

Let $\Sigma = \{a_1, \ldots, a_s\}$ and $\Gamma = \{b_1, \ldots, b_t\}$, ordered by $b_1 < \cdots < b_t$. Let furthermore $k > 0$ and let $\overline{x}$ be a k-tuple of variables (each of which ranges from 1 to the "input length" n, as we have seen). In the following, we assume the lexical ordering on $\{1, 2, \ldots, n\}^k$, and we write $X_1, X_2, \ldots, X_{n^k}$ for the sequence of potential values taken on by $\overline{x}$. Let $\phi_1, \ldots, \phi_{t-1}$ be first-order formulas over Σ, each with free variables $\overline{x}$. These formulas define a *transformation* $[\phi_1, \ldots, \phi_{t-1}] \colon \Sigma^* \to \Gamma^*$ as follows: Let $w = w_1 \cdots w_n \in \Sigma^*$. Then, $[\phi_1, \ldots, \phi_{t-1}](w) = v_1 \cdots v_{n^k} \in \Gamma^*$, where

$$v_i = \begin{cases} b_1 & \text{if } w \models \phi_1(X_i), \\ b_2 & \text{if } w \models \neg\phi_1(X_i) \wedge \phi_2(X_i), \\ \vdots & \qquad\qquad \vdots \\ b_{t-1} & \text{if } w \models \neg\phi_1(X_i) \wedge \neg\phi_2(X_i) \wedge \ldots \wedge \phi_{t-1}(X_i), \\ b_t & \text{if } w \models \neg\phi_1(X_i) \wedge \neg\phi_2(X_i) \wedge \ldots \wedge \neg\phi_{t-1}(X_i), \end{cases}$$

for $1 \leq i \leq n^k$. In the case that $k = 1$, the transformation is also referred to as a *translation*.

Note that, in the definition of a transformation, we make use of $t-1$ formulas if the target alphabet consists of t symbols. The transformation takes the t-th symbol whenever all $t-1$ formulas are false. Transformations could be defined differently. We chose this way to assure that each sequence of formulas defines a total function on every input string.

Definition 2.1. Consider a language L over an alphabet $\Gamma = (b_1, b_2, \ldots, b_t)$. Such a language gives rise to a *Lindström quantifier* Q_L that may be used in combination with a transformation $[\phi_1, \ldots, \phi_{t-1}]$ as follows. The formula

$$Q_L\overline{x}\left[\phi_1(\overline{x}), \phi_2(\overline{x}), \ldots, \phi_{s-1}(\overline{x})\right]$$

holds on a string $w = w_1 \cdots w_n$ over Σ if $[\phi_1, \ldots, \phi_{t-1}](w) \in L$. In case $\Sigma = \{0, 1\}$ ($s = 2$) we omit the braces and write $Q_L\overline{x}\phi(\overline{x})$ for short.

The Lindström quantifiers of Definition 2.1 are more precisely what has been referred to as "Lindström quantifiers on strings" [BV98]. The original more general definition [Lin66] uses transformations to arbitrary structures, not necessarily of string signature. However, in the context of this article, only reductions to languages such as CFLs or algebraic word problems will be important, and hence the above definition seems to be the most natural here.

2.3 Groupoid-based language classes

A *groupoid* is a set G with a binary operation. No constraint such as associativity or the existence of an identity is assumed. Hence, though this is not apparent from the terminology (rousing the connotation of a *group*), every monoid is a groupoid.

Fix a finite groupoid G. Each $S \subseteq G$ defines a G-word problem, i.e., a language $\mathcal{W}(G, S)$ composed of all words w, over the alphabet G, that "multiply out" to an element of S when an appropriate legal bracketing of w is chosen.

Definition 2.2. A *groupoidal quantifier* is a Lindström quantifier Q_L where L is a word problem of some finite groupoid.

A folklore result, generally credited to Valiant (see [BLM93, Lemma 3.1]) states that any context-free language is a word problem over some groupoid, and vice-versa every word problem of a finite groupoid is context-free. Hence, a groupoidal quantifier is nothing other than a Lindström quantifier Q_L where L is a context-free language.

Coming back for a moment to the classical definition of Lindström quantifiers, we thus see that a Lindström quantifier on strings defined by a context-free language is nothing other than a Lindström quantifier (in the classical sense) defined by a structure that is a finite groupoid multiplication table.

When Barrington, Immerman, and Straubing defined *monoidal quantifiers* in [BIS90], they proceeded along the same avenue: they first showed how monoid word problems can be seen as languages, and then defined generalized quantifiers given by such languages (see [BIS90, pp. 284f.]).

Definition 2.3. Q_GFO is the set of languages describable by a formula $Q_L[\phi_1, \ldots, \phi_t]$, where $L = \mathcal{W}(G, S)$ for some $S \subseteq G$.

Q_{Grp}FO is the union, over all finite groupoids G, of Q_GFO.

FO(Q_G) and FO(Q_{Grp}) are defined analogously, but allowing groupoidal quantifiers to be used as any other quantifier would (i. e., allowing arbitrary nesting).

Q_G^{un}FO and FO($Q_{\mathrm{Grp}}^{\mathrm{un}}$), etc, are defined analogously, but restricting to *unary* groupoidal quantifiers (signaled by the exponent "un").

$Q_G\mathrm{FO}(+,\times)$, $Q_G^{\mathrm{un}}\mathrm{FO}(+,\times)$, $Q_{\mathrm{Grp}}\mathrm{FO}(+,\times)$, $\mathrm{FO}(+,\times,Q_G)$, etc., are defined analogously, but allowing the arithmetical predicates plus and times.

Examples. The FO-formula $P_1(\min) \wedge P_0(\max) \wedge \forall x \forall y \big(\big((x < y) \wedge \neg \exists z (x < z \wedge z < y)\big) \to \big((P_0(x) \to P_1(y)) \wedge (P_1(x) \to P_0(y))\big)\big)$ defines the language $(10)^+$. In fact, is is known that FO is precisely the class of all star-free regular sets [MP71]. From this, it follows that $\{\, w \mid |w| \equiv 0 \pmod 2 \,\}$ is not in FO, because it is not star-free; on the other hand it is easily seen to be in FO$(+,\times)$, even in FO$(+)$.

The class FO$(+,\times)$ is known to be equal to uniform AC^0 [Imm89], hence it also contains non-regular sets such as $\{\, w \mid (\exists n)(|w| = n^2) \,\}$ and $\{\, w \mid (\exists n)(|w| = 2^n) \,\}$. In Subsect. 4.3.2 we will see that these latter languages cannot even be expressed in first-order logic with unary groupoidal quantifiers (i.e., they are not in FO($Q_{\mathrm{Grp}}^{\mathrm{un}}$)). In fact, every one-letter-alphabet language that is non-regular will turn out not to be in FO($Q_{\mathrm{Grp}}^{\mathrm{un}}$). On the other hand, there are regular sets, e.g., the word problem $\mathcal{W}(S_5, \{1\})$ for the symmetric group S_5 (with identity 1), that are complete for NC^1 (under AC^0-reductions) [Bar89], hence not in AC^0.

The quantifier $\exists$ is the quantifier $Q_{\mathcal{W}(OR,\{1\})}$, where OR is the aperiodic monoid defined by the binary OR. Straubing [Str94] surveys an elegant theory in which FO is supplemented with "monoidal" quantifiers such as MOD_q, where $\mathrm{MOD}_q x\, \varphi$ holds iff $\varphi(x)$ holds in a multiple of q word positions x. Such quantifiers are $Q_{\mathcal{W}(\mathbb{Z}_q,\{0\})}$ quantifiers. Hence $[Q_{\mathcal{W}(\mathbb{Z}_q,\{0\})} y (y < x \wedge C_a y)]$ is a translation, mapping $w_1 w_2 \cdots w_n \in \{a,b,c\}^*$ to $v_1 \cdots v_n \in \{0,1\}^n$, such that $v_i = 1$ iff $w_1 \cdots w_{i-1}$ contains a multiple of q occurrences of a.

2.4 Unary quantifiers and homomorphisms

We will encounter unary groupoidal quantifiers repeatedly. Here we show how these relate to standard formal language operations. Recall that a length-preserving homomorphism $\Sigma^* \to \Delta^*$ is the unique free monoid morphism extending a map $h\colon \Sigma \to \Delta$ for finite alphabets Σ, Δ. In a different context, a result very similar to the next theorem is known as *Nivat's Theorem* [MS97, Theorem 3.8, p. 207].

Theorem 2.4. *Let B be an arbitrary language, and let A be describable in $Q_B^{\mathrm{un}}\mathrm{FO}$, that is, by a first order sentence preceded by one unary Lindström quantifier. Then there are length-preserving homomorphisms g, h and a regular language D such that $A = h(D \cap g^{-1}(B))$.*

Proof. Let $\Sigma = \{a_1, \ldots, a_s\}$ and let $A \subseteq \Sigma^*$ be defined by the formula $\psi \in Q_B^{\mathrm{un}}\mathrm{FO}$, $\psi = Q_B x \phi(x)$, $B \subseteq \Gamma^*$. For convenience, we assume $\Gamma = (0, 1)$. The general case is analogous. Formula ϕ thus defines a translation from words over Σ to binary words. Define $D \subseteq (\Sigma \times \Gamma)^*$ to consist of all words $\left[\begin{smallmatrix} u_1 \\ y_1 \end{smallmatrix}\right] \cdots \left[\begin{smallmatrix} u_k \\ y_k \end{smallmatrix}\right]$ such that ϕ maps $u_1 \cdots u_k$ to $y_1 \cdots y_k$. Define the homomorphisms h and g by $h\colon \left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right] \mapsto a$ and $g\colon \left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right] \mapsto b$ for all $a \in \Sigma$ and $b \in \Gamma$. Then $h(D \cap g^{-1}(B)) = A$.

It remains to show that D is regular. Let $\phi'(x)$ be obtained from ϕ by replacing, for each symbol $\sigma \in \Sigma$ and each variable z, each term $P_\sigma(z)$ by $P_{\left[\begin{smallmatrix} \sigma \\ 0 \end{smallmatrix}\right]}(z) \vee P_{\left[\begin{smallmatrix} \sigma \\ 1 \end{smallmatrix}\right]}(z)$. Hence, for each $i \leq |w|$, $\phi'(i)$ holds on a string w over $\Sigma \times \Gamma$, if and only if $\phi(i)$ holds on $h(w)$. Let $\psi(x)$ be the formula $P_{\left[\begin{smallmatrix} a_1 \\ 1 \end{smallmatrix}\right]}(z) \vee \cdots \vee P_{\left[\begin{smallmatrix} a_s \\ 1 \end{smallmatrix}\right]}(z)$. Then a string $v \in (\Sigma \times \Gamma)^*$ is in D if and only if $\forall x\ \phi'(x) \leftrightarrow \psi(x)$ holds on v. Therefore, D is regular. ❑

Remark 2.5. Since FO precisely captures the variety of star-free regular languages [MP71], we even conclude that the D above is star-free.

3 An automaton characterization of FO-translations

As a technical tool, it will be convenient to have an automata-theoretic characterization of *first-order translations*, i.e., of reductions defined by FO-formulas with one free variable. Since FO precisely describes the (regular) languages accepted by aperiodic deterministic finite automata [MP71], one might expect aperiodic deterministic finite transducers to capture FO-translations. This is not the case however because, e.g. the FO-translation which maps every string $w_1 \cdots w_n$ to w_n^n cannot be computed by such a device.

We show in this section that the appropriate automaton model to use is that of a single-valued aperiodic nondeterministic finite transducer, which we define and associate with FO-translations in this section. But first, we discuss the notion of an aperiodic NFA.

Definition 3.1. A deterministic or nondeterministic FA M is *aperiodic* (or group-free) iff there is an $n \in \mathbb{N}$ such that for all states s and all words w,

$$\delta(s, w^n) = \delta(s, w^{n+1}).$$

Here δ is the extension of M's transition function from symbols to words. Observe that if M is nondeterministic then $\delta(t, v)$ is a set of states, i.e., locally here we abuse notation by not distinguishing between M's extended transition function δ and the function δ^* as defined in the context of a nondeterministic transducer below.

Remark 3.2. This definition of aperiodicity for a DFA is the usual one (see [Ste85]). For a NFA, a statement obviously equivalent to Definition 3.1 would be that A is aperiodic iff applying the subset construction to A yields an aperiodic DFA. Hence [Sch65] a language L is star-free iff some aperiodic (deterministic or nondeterministic) finite automaton accepts L.

We now prepare the ground for the main result of this section, namely that single-valued aperiodic nondeterministic finite transducers characterize FO-translations.

Definition 3.3. A *finite transducer* is given by a set Q of *states*, an *input alphabet* Σ, an *output alphabet* Γ, an *initial state* q_0, a *transition relation* $\delta \subseteq Q \times \Sigma \times \Gamma \times Q$ and a set $F \subseteq Q$ of *final states.* For a string $w = w_1 \cdots w_n \in \Sigma^*$ we define the *set* $O_M(w)$ *of outputs of* M *on input* w as follows. A string $v \in \Gamma^*$ of length n is in $O_M(w)$, if there is a sequence $s_0 = q_0, s_1, \dots, s_n$ of states, such that $s_n \in F$ and, for every i, $1 \leq i \leq n$, we have $(s_{i-1}, w_i, v_i, s_i) \in \delta$ (where we let $s_0 = s$).

We say that M is *single-valued* if, for every $w \in \Sigma^*$, $|O_M(w)| = 1$. If M is single-valued it naturally defines a function $f_M : \Sigma^* \to \Gamma^*$.

For every string $u \in \Sigma^*$ and every state $s \in Q$ we write $\delta^*(s, u)$ for the set of states s' that are reachable from s on input u (i. e., there are $s_1, \dots, s_{|u|} = s'$ and $v_1 \cdots v_{|u|}$ such that, for every i, $1 \leq i \leq |u|$, we have $(s_{i-1}, u_i, v_i, s_i) \in \delta$).

As per Definition 3.1, M is *aperiodic* if there is an $n \in \mathbb{N}$ such that for all states q and all strings w, $\delta^*(q, w^n) = \delta^*(q, w^{n+1})$.

We will need some basic properties of FO logic on strings.

Let k be a fixed natural number and Σ an alphabet. For every string u we write Φ_u^k for the set of FO-sentences of quantifier-depth k that hold in u. Let S^k denote the set $\{\Phi_u^k \mid u \in \Sigma^*\}$. It is well-known that, modulo logical equivalence, S^k is finite for every fixed k and Σ, see [EFT94, p. 253].

Lemma 3.4. *Let u, u', v, v' be strings such that $\Phi_u^k = \Phi_{u'}^k$ and $\Phi_v^k = \Phi_{v'}^k$. Then $\Phi_{uv}^k = \Phi_{u'v'}^k$.*

Proof. As $\Phi_u^k = \Phi_{u'}^k$ and $\Phi_v^k = \Phi_{v'}^k$ we know that the duplicator has a winning strategy in the k-round Ehrenfeucht game (see [EFT94] for an introduction to Ehrenfeucht games) on u and u' and in the game on v and v'. These strategies can be easily combined to get a winning strategy on uv and $u'v'$. From the existence of this winning strategy we can, in turn, conclude that $\Phi_{uv}^k = \Phi_{u'v'}^k$. ❑

Theorem 3.5. *A function $f\colon \Sigma^* \to \Gamma^*$ is defined by an* FO *translation if and only if it is defined by a single-valued aperiodic finite transducer.*

Proof. To simplify notation we assume that $\Gamma = (0,1)$. The proof of the general case is a straightforward generalization.

(only if) Let $f\colon \Sigma^* \to \Gamma^*$ be defined by formula $\varphi(x)$ of quantifier-depth k (hence, for every $w \in \Sigma^*$ and every $i \le |w|$, the i-th bit of $f(w)$ is 1 iff $w \models \varphi(i)$). We define a single-valued aperiodic finite transducer M with input alphabet Σ, output alphabet Γ, set $S^k \times S^k \cup \{q_0\}$ of states, initial state q_0 and accepting states $\{(\Phi, \Phi^k_\epsilon) \mid \Phi \in S^k\}$. Informally, a state (Φ_1, Φ_2) of M represents a situation, in which M "knows" that Φ_1 contains exactly those formulas (of quantifier depth k) that hold in the prefix of the input string that was already read, and it "guesses" that Φ_2 contains exactly those formulas that hold in the remaining part of the string.

The transition relation δ of M is defined as follows. For every $\Phi_1, \Phi_2, \Phi'_1, \Phi'_2 \in S^k$, every $\sigma \in \Sigma$ and every $\tau \in \Gamma$ we let

$$((\Phi_1, \Phi_2), \sigma, \tau, (\Phi'_1, \Phi'_2)) \in \delta,$$

if there exist strings $u, v \in \Sigma^*$ such that $\Phi_1 = \Phi^k_u$, $\Phi'_1 = \Phi^k_{u\sigma}$, $\Phi_2 = \Phi^k_{\sigma v}$, $\Phi'_2 = \Phi^k_v$ and $\tau = 1 \iff u\sigma v \models \varphi(|u|+1)$.

Analogously, for every $\Phi'_1, \Phi'_2 \in S^k$, every $\sigma \in \Sigma$ and every $\tau \in \Gamma$ we define

$$(q_0, \sigma, \tau, (\Phi'_1, \Phi'_2)) \in \delta,$$

if there exists a string $v \in \Sigma^*$ such that $\Phi'_1 = \Phi^k_\sigma$, $\Phi'_2 = \Phi^k_v$ and $\tau = 1 \iff \sigma v \models \varphi(1)$.

We first check that M is single-valued. Let $w = w_1 \cdots w_n$, and $f(w) = v_1 \cdots v_n$. We set $s_0 = q_0$ and, for every $i > 0$, $s_i = (\Phi^k_{w_1 \cdots w_i}, \Phi^k_{w_{i+1} \cdots w_n})$. By using Lemma 3.4, it is easy to verify that $s_n \in F$ and, for every $i > 0$, we have $(s_{i-1}, w_i, v_i, s_i) \in \delta$. Hence $f(w) \in O_M(w)$.

We have to show now that no string $u = u_1 \cdots u_n \ne f(w)$ is in $O_M(w)$. Assume otherwise and let $s'_0 = q_0, s'_1, \ldots, s'_n$ be a sequence of states that outputs u. Let, for every $i > 0$, $s'_i =: (\Psi_i, \Theta_i)$. First, it is easy to observe that, for every $i > 0$, $\Psi_i = \Phi^k_{w_1 \cdots w_i}$. As u is different from v there must be a j such that $\Theta_j \ne \Phi^k_{w_{j+1} \cdots w_n}$ (note that from the definition of δ it follows that $(s, \sigma, 1, s') \in \delta$ implies $(s, \sigma, 0, s') \notin \delta$). We conclude that for every $i > j$, $\Theta_i \ne \Phi^k_{w_{i+1} \cdots w_n}$: Assume, otherwise that $i > j$ is minimal, such that $\Theta_i = \Phi^k_{w_{i+1} \cdots w_n}$. By definition of δ and as $(s'_{i-1}, w_i, \tau, s'_i) \in \delta$ it follows immediately that $\Theta_{i-1} = \Phi^k_{w_i w_{i+1} \cdots w_n}$, a contradiction. Hence, in particular, $\Theta_n \ne \Phi^k_\epsilon$, i.e., $s'_n \notin F$. It follows that M is single-valued and $f_M = f$.

It remains to show that M is aperiodic. First of all, it is well-known, and can be shown by an Ehrenfeucht game argument (see [EF95]) that, for $n = 2^k$ and every $w \in \Sigma^*$, we have $\Phi^k_{w^n} = \Phi^k_{w^{n+1}}$.

Let now $\Phi_1, \Phi_2, \Phi'_1, \Phi'_2 \in S^k$ and let $u, v \in \Sigma^*$ with $\Phi_1 = \Phi^k_u$ and $\Phi'_2 = \Phi^k_v$. From Lemma 3.4 and the definition of δ we can conclude that $(\Phi'_1, \Phi'_2) \in \delta^*((\Phi_1, \Phi_2), x)$

if and only if $\Phi_2 = \Phi^k_{xv}$ and $\Phi'_1 = \Phi^k_{ux}$. Hence, again with Lemma 3.4, we get for every w the following.

$$\begin{aligned}(\Phi'_1, \Phi'_2) \in \delta^*((\Phi_1, \Phi_2), w^n) &\iff \Phi_2 = \Phi^k_{w^n v} \text{ and } \Phi'_1 = \Phi^k_{uw^n} \\ &\iff \Phi_2 = \Phi^k_{w^{n+1} v} \text{ and } \Phi'_1 = \Phi^k_{uw^{n+1}} \\ &\iff (\Phi'_1, \Phi'_2) \in \delta^*((\Phi_1, \Phi_2), w^{n+1})\end{aligned}$$

This implies that M is aperiodic.

(if) Let f be computed by a single-valued aperiodic finite transducer $M = (Q, \Sigma, \Gamma, q_0, \delta, F)$. It is easy to check that, for every $s, s' \subseteq Q$, the language

$$L(s, s') = \{u \mid s' \in \delta^*(s, u)\}$$

is accepted by an aperiodic finite automaton. Consequently, every $L(s, s')$ is characterized by a FO formula $\varphi^{s,s'}$. Let $\varphi(x)$ be the formula

$$\bigvee_{\substack{s,s',s'',\sigma \\ s'' \in F \wedge (s,\sigma,1,s') \in \delta}} \varphi^{q_0,s}_{<}(x) \wedge P_\sigma(x) \wedge \varphi^{s',s''}_{>}(x).$$

Here, for every s and s', $\varphi^{s,s'}_{<}(x)$ is the formula that is obtained by relativizing $\varphi^{s,s'}$ to all positions that are smaller than x and $\varphi^{s,s'}_{>}(x)$ is the formula that is obtained by relativizing $\varphi^{s,s'}$ to all positions that are greater than x (see for example [Str94, pp. 81f]).

Hence, for every position x, $\varphi(x)$ becomes true in a string w if and only if there are states s, s', s'' such that

- M can reach s from the initial state by reading the string to the left of x,
- M can reach s' from s by reading the symbol at position x and output a 1, and
- M can reach the final state s'' from s' by reading the string to the right of x.

As M is single-valued, $\varphi(x)$ defines $f_M(w)$, for every w. ❑

4 First-order with groupoidal quantifiers

4.1 The largest attainable class: LOGCFL

Theorem 4.1. *There is a fixed groupoid G such that*

$$Q_G\mathrm{FO}(+, \times) = \mathrm{FO}(+, \times, Q_{\mathrm{Grp}}) = \mathrm{LOGCFL}.$$

Proof. $Q_G\mathrm{FO}(+,\times) \subseteq \mathrm{FO}(+,\times,Q_{\mathrm{Grp}})$ holds by definition for any groupoid G. To see that $\mathrm{FO}(+,\times,Q_{\mathrm{Grp}}) \subseteq \mathrm{LOGCFL}$, note that [BIS90, Theorem 8.1] implies the existence of a logspace-uniform AC^0-reduction, from any language in $\mathrm{FO}(+,\times,Q_{\mathrm{Grp}}) = \mathrm{FO}(\mathrm{BIT},Q_{\mathrm{Grp}})$, to a set of groupoid word problems. The unbounded fan-in AND gates in the AC^0 reduction can be replaced by log depth bounded fan-in subcircuits. Then the groupoid word problem oracle gates, of which no more than a constant number can appear on any path from circuit inputs to circuit output, can be expanded into SAC^1 subcircuits, since groupoid word problems are context-free languages. There results a logspace-uniform SAC^1 circuit, proving membership in LOGCFL.

$\mathrm{LOGCFL} \subseteq Q_G\mathrm{FO}(+,\times)$ is seen by appealing to the fixed G whose word problem is LOGCFL-complete under DLOGTIME reducibility [BLM93]. Since DLOGTIME was shown expressible in $\mathrm{FO}(\mathrm{BIT}) = \mathrm{FO}(+,\times)$ by [BIS90], the inclusion follows. ❑

4.2 Capturing LOGCFL without arithmetic

Theorem 4.2. *There is a fixed groupoid G such that* $\mathrm{LOGCFL} \subseteq Q_G\mathrm{FO}$.

Proof. We first show how to express plus and times and their negations as $\mathrm{FO}^+(Q_{\mathrm{Grp}})$ formulas (i.e., formulas which have outside of the groupoidal quantifier only a first-order quantifier prefix and in particular no negation).

Let us look at the predicate "$a \times b = c$." Let

- $L =_{\mathrm{def}} \{\, w \in (0,1,\#)^* \mid |w|_0 = |w|_1 \,\}$,
- $\varphi_1 =_{\mathrm{def}} (z = \min) \wedge (x \leq a) \wedge (y \leq b)$,
- $\varphi_2 =_{\mathrm{def}} (z = y = \max) \wedge (x \leq c)$,

and finally define
$$\phi(a,b,c) =_{\mathrm{def}} Q_L(x,y,z)\,[\varphi_1,\varphi_2]\,.$$
Given a word w of length n and assignments for a,b,c, the transformation $[\varphi_1,\varphi_2]$ yields a string of length n^3 over the alphabet $(0,1,\#)$ which contains $a\cdot b$ many 0s and c many 1s. The remaining positions carry the symbol $\#$. Thus this image is in L if and only if $a \cdot b = c$. Observe that L is deterministic context-free, therefore its complement is context-free and we conclude that we can also express $a \times b \neq c$ by a $\mathrm{FO}^+(Q_{\mathrm{Grp}}\mathrm{FO})$ formula (in fact even by a $Q_{\mathrm{Grp}}\mathrm{FO}$ formula).

In a similar (even simpler) way we can express $a + b = c$ and $a + b \neq c$ by $\mathrm{FO}^+(Q_{\mathrm{Grp}})$ formulas. Again, all context-free languages involved in the definition of these predicates are context-free and co-context-free. Furthermore, if we choose disjoint alphabets for these languages, their union L_0 is again context-free and co-context-free and all the mentioned predicates can be expressed by using the quantifier Q_{L_0}.

Let now $A \in$ LOGCFL be fixed. From Theorem 4.1 we know that LOGCFL $= Q_{\mathrm{Grp}}\mathrm{FO}(+,\times)$. Thus A can be defined by a formula

$$Q_L\overline{x}\big[\Phi_1,\ldots,\Phi_{s-1}\big], \tag{1}$$

where each Φ_i is a FO$(+,\times)$ formula and L is context-free over some alphabet $\Sigma = (a_1,\ldots,a_s)$.

In the following we will show how every such formula can be transformed into a $Q_{L'}$FO-formula, for some fixed context-free language L'.

Using the argument above we can replace each Φ_i in (1) by a formula without plus and times but using a Q_{L_0} quantifier, where L_0 is context-free and co-context-free. This formula can then be transformed into the form

$$\exists\overline{x}_1\forall\overline{x}_2\exists\overline{x}_3\cdots\bigvee_{i_1}\bigwedge_{i_2}\phi_{i_1,i_2}, \tag{2}$$

where each of the ϕ_{i_1,i_2} is either a positive atomic formula or a formula of the form $Q_{L_0}\chi$, where χ is quantifier-free and L_0 is context-free and co-context-free.

Now we combine stepwise the inner quantifiers Q_{L_0} $(1 \le j \le m)$ in formula (2) with the first-order connectives $\bigvee$, $\bigwedge$ and the first-order quantifiers $\exists$, $\forall$. We give the construction for the case of an existential quantifier. Consider the formula $\exists x Q_{L_1}\overline{y}\big[\xi_1,\ldots,\xi_{k-1}\big]$, where $L_1 \subseteq \Gamma^*$ is context-free and co-context-free. Suppose $\Gamma = (a_1,\ldots,a_k)$, $\# \notin \Gamma$. Let $\overline{y} = (y_1,\ldots,y_l)$. This formula is equivalent to $Q_{L_2}(x,z,y_1,\ldots,y_l)\big[\xi_0,\xi'_1,\ldots,\xi'_{k-1}\big]$ where

$$L_2 = \big\{\, w \in (\#,a_1,\ldots,a_k)^* \;\big|\; w = w_1\#^+w_2\#^+\cdots\#^+w_n\#^+,\ w_i \in L_1 \text{ for some } i \,\big\},$$

ξ_0 is the formula $z > 1$ and each ξ'_i, $1 \le i \le k-1$, is defined as $(z = 1) \wedge \xi_i$. The transformation f defined by $[\xi_0,\xi'_1,\ldots,\xi'_{k-1}]$ maps a word w of length n to a word $f(w)$ of length n^{l+2}. $f(w)$ can be viewed as consisting of n blocks $u_1,\ldots,u_n$ of length n^{l+1} each, where each u_m corresponds to $x = m$ and itself consists of n blocks of length n^l, one block for each value of z. These blocks are all in $\#^*$ for $z > 1$, and consist of a word over Γ for $z = 1$. This word is exactly the word to which w is mapped under the transformation $[\xi_1,\ldots,\xi_{k-1}]$, when $x = m$. Hence we see that $f(w) \in L_2$ if there is some value m such that $u_m \in L_1\#^*$. This proves the correctness of the above construction. Certainly L_2 is context-free, and since the complement of L_1 is context-free, we see that the complement of L_2 is also context-free (the construction of appropriate PDAs is obvious).

The combinations of a Q_{L_j} with a universal quantifier, or with a first-order connective, are dealt with analogously.

We thus replaced the subformulas Φ_i in formula (1) above and obtained a formula of the form

$$Q_L\overline{x}\big[\Psi_1,\ldots,\Psi_{s-1}\big], \tag{3}$$

where, for each i, Ψ_i is of the form $Q_{L_i}\overline{y}[\psi^i_1,\ldots,\psi^i_{l_i}]$, for some l_i, where all ψ^i_j are quantifier-free and L_i is context-free and co-context-free. Let Δ be an alphabet which contains all symbols occuring in the laguages $L_1, \dots, L_{s-1}$ and let $\#, \$$ be new symbols, not contained in Δ. We now define a substitution h by

$$\begin{aligned}
h(a_1) &= \$L_1\#^+(\Delta^*\#^+)^{s-2} \\
h(a_2) &= \$\overline{L_1}\#^+L_2\#^+(\Delta^*\#^+)^{s-3} \\
&\cdots \\
h(a_{s-1}) &= \$\overline{L_1}\#^+\cdots\#^+\overline{L_{s-2}}\#^+L_{s-1}\#^+ \\
h(a_s) &= \$\overline{L_1}\#^+\overline{L_2}\#^+\cdots\#^+\overline{L_{s-1}}\#^+
\end{aligned}$$

and let $L' =_{\rm def} h(L)$.

Now, for each i, we have a transformation $\tau_i = [\psi_1^i, \dots, \psi_{l_i}^i]$ which produces on input w, for each fixed choice $\overline{q}$ for the variables $\overline{x}$ a string $w_i^{\overline{q}}$. By construction of h we can conclude that w is in A, if and only if a string of the form

$$\$w_1^{X(1)}\#^+w_2^{X(1)}\#^+\cdots w_{s-1}^{X(1)}\#^+\cdots\$w_1^{X(n^k)}\#^+w_2^{X(n^k)}\#^+\cdots w_{s-1}^{X(n^k)}\#^+ \qquad (4)$$

is in $h(L)$, where $X(1), X(2), \dots, X(n^k)$ are the possible values for $\overline{x} = x_1, \dots, x_k$ in lexicographic order. It should be noted that all strings of the form (4) have the same behaviour with respect to $h(L)$.

It is now straightforward, though tedious, to define a quantifier-free transformation $[\chi_1, \dots, \chi_m]$ (where $m = |\Delta| + 1$) which produces a word of the form (4) on input w. Hence, our formula replacing (1) can be chosen as

$$Q_{L'}\overline{z}[\chi_1, \dots, \chi_m], \qquad (5)$$

where all formulas χ_i are quantifier-free.

Thus we have shown that LOGCFL $\subseteq Q_{\rm Grp}$FO. Now define H to be Greibach's hardest context-free language. Any CFL L reduces to H via a homomorphism (see [Har78, p. 326]. This homomorphism is ϵ-free but not length-preserving. Applying a non-unary groupoidal quantifier to simple FO-formulas can realize this homomorphism, provided that a new padding or neutral symbol be introduced, to act as a filler in any word. Thus we see that any Q_LFO formula can be transformed into an equivalent $Q_{{\rm pad}(H)}$FO formula. ❑

A corollary to this proof is the following remarkable result:

Corollary 4.3. *Greibach's hardest context-free language with a neutral symbol is complete for* LOGCFL *under quantifier-free reductions without arithmetic.*

A noteworthy strengthening of Theorem 4.1 thus follows from Theorem 4.2:

Corollary 4.4. *There is a fixed groupoid G such that* Q_GFO $=$ FO(Q_G) $=$ LOGCFL.

4.3 Unary groupoidal quantifiers

In the previous subsection, we have shown that the situation with non-unary groupoidal quantifiers is clearcut, since a single such quantifier, even without the arithmetical predicates (or, without BIT), captures all of LOGCFL. Here we examine the case of unary quantifiers. In this case, the presence or absence of the arithmetical predicates is once again relevant.

4.3.1 Unary groupoidal quantifiers without arithmetic

Theorem 4.5. $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO} = \mathrm{CFL}$.

Proof. The direction from right to left follows from [BLM93]: Every context-free language reduces via a length-preserving homomorphism to a groupoid word problem. We can even look at the letters in a given word as groupoid elements. This reduction can be expressed in FO.

The direction from left to right is proved by appealing to Theorem 2.4 and observing that the context-free languages have the required closure properties. ❑

It follows immediately that nesting unary groupoidal quantifiers (in fact, merely taking the Boolean closure of $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}$) adds expressiveness. Let us write $\mathrm{BC}(\mathcal{L})$ to denote the Boolean closure of the set $\mathcal{L}$ of languages (i.e., closure under intersection, union, and complement) and $\mathrm{BC}^+(\mathcal{L})$ to denote the closure under union and intersection only.

Corollary 4.6.

$$\begin{aligned} Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO} = \mathrm{CFL} &\subsetneq \mathrm{BC}^+(Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}) = \mathrm{BC}^+(\mathrm{CFL}) \\ &\subsetneq \mathrm{BC}(Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}) = \mathrm{BC}(\mathrm{CFL}) \\ &\subseteq \mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}}). \end{aligned}$$

Proof. All inclusions from left to right are clear. The first separation follows from the fact that CFLs are not closed under intersection. The second separation follows from considering the non-context-free language Y, consisting of all words of the form ww, the complement of which is context-free. ❑

The inclusion $\mathrm{CFL} \subseteq Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}$ in Theorem 4.5 could have be proved alternatively by observing that the logic $\exists M\mathrm{FO}$ capturing CFL (see [LST94]) is closed under FO translations. We note in the same vein:

Theorem 4.7. $Q_{\mathrm{Grp}}\mathrm{SOM} = \mathrm{CFL}$.

Proof. In [LST94] it is in fact proved that $\mathrm{CFL} = \exists M\mathrm{SOM}$. This logic is closed under monadic second-order (SOM) transformations. Hence $\mathrm{CFL} \subseteq Q_{\mathrm{Grp}}\mathrm{SOM} \subseteq \exists M\mathrm{SOM} \subseteq \mathrm{CFL}$. ❑

Can we refine Theorem 4.5 and find a universal finite groupoid G that captures all the context-free languages as Q_G^{un}FO? Intuition from the world of monoids [BIS90, p. 303] suggests that the answer is no. Proving the nonexistence of G is the content of Theorem 4.9 below. We first make a definition and state a lemma.

Let D_t be the context-free one-sided Dyck language over $2t$ symbols, i.e., D_t consists of the well-bracketed words over an alphabet of t distinct types of parentheses. Recall that a PDA is a nondeterministic automaton that reads its input from left to right and has access to a pushdown store with a fixed pushdown alphabet. We say that a PDA A is *k-pushdown-limited*, for k a positive integer, iff

- the pushdown alphabet of A has size k, and
- A pushes no more than k symbols on its stack between any two successive input head motions.

Lemma 4.8. *No k-pushdown-limited PDA accepts D_t when $t \geq (k+1)^k + 1$.*

Proof. Suppose to the contrary that a k-pushdown-limited PDA A accepts D_t, where $t = (k+1)^k + 1$. A has a certain fixed number, s, of states. Let us call a state of A together with a stack content of A a *situation*. Consider A's computation as it scans a length-n prefix of its input. Since A is k-pushdown-limited, no more than $(k+1)^{kn}$ different stack contents, hence no more than $s \cdot (k+1)^{kn}$ situations are encountered. But A must be able to distinguish between each pair of length-n prefixes consisting of left parentheses alone, because for any two such prefixes v_1 and v_2, there is a Dyck word $v_1 w$ such that $v_2 w$ is not a Dyck word. Now, it is easy to see that t^n, the number of length-n words over an alphabet of t left parentheses, exceeds $s \cdot (k+1)^{kn}$ when n is large. Hence A cannot accept D_t. ❑

Theorem 4.9. *For any finite groupoid G, Q_G^{un}FO $\subsetneq$ CFL.*

Proof. Suppose to the contrary that G is a finite groupoid such that Q_G^{un}FO = CFL. Then there is a FO-translation from each context-free language to a word problem for G. This means that a finite set of PDAs (one for each word problem $\mathcal{W}(G, \cdot)$) can take care of answering each "oracle question" resulting from such a FO-translation. By Theorem 3.5, each FO-translation is computed by a single-valued NFA. Although the NFAs differ for different context-free languages (and this holds in particular when language alphabets differ), the NFAs do not bolster the "pushdown-limits" of the PDAs which answer all oracle questions. Hence if k is a fixed integer such that all word problems $\mathcal{W}(G, \cdot)$ for G are accepted by a k-pushdown-limited PDA, then for any positive integer t, D_t is accepted by a k-limited-pushdown PDA. This contradicts Lemma 4.8 when $t = (k+1)^k + 1$. ❑

In the next subsection we will see that the arithmetical predicates provably add power to the logic $Q_{\text{Grp}}^{\text{un}}$FO. Since it is known, as we mentioned in the preliminaries, that the arithmetical predicates and the BIT predicate are equally expressive, and the latter can be expressed by the majority of pairs quantifier [BIS90], i.e., the majority quantifier binding two variables, the following two simple observations about the power of $Q_{\text{Grp}}^{\text{un}}$FO are of particular interest.

Theorem 4.10. *The (unary) majority quantifier is definable in* $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}$.

Proof. Majority is a context-free language. ❑

Theorem 4.11. *Addition is definable in* $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}$.

Proof. Let i, j, k be positions in the input word. We want to express that $i + j = k$. We do this by using a quantifier for the context-free language $L =_{\mathrm{def}} \left\{ 0^{i-1}a1^*b0^{i-1}c1^* \mid i \in \mathbb{N} \right\}$. Given a word $w \in L$, if symbol a is at position i and b is at position j, then c must be at position $i + j$. ❑

4.3.2 Unary groupoidal quantifiers with arithmetic

What are $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}(+,\times)$ and $\mathrm{FO}(+,\times,Q^{\mathrm{un}}_{\mathrm{Grp}})$? It would seem plausible that $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}(+,\times) \subsetneq \mathrm{FO}(+,\times,Q^{\mathrm{un}}_{\mathrm{Grp}}) \subsetneq \mathrm{LOGCFL}$, but we are unable to prove $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}(+,\times) \subsetneq \mathrm{LOGCFL}$, much less $\mathrm{FO}(+,\times,Q^{\mathrm{un}}_{\mathrm{Grp}}) \subsetneq \mathrm{LOGCFL}$. The next lemma indicates that proving the latter would prove $\mathrm{TC}^0 \neq \mathrm{LOGCFL}$, settling a major open question in complexity theory.

Lemma 4.12. $\mathrm{TC}^0 \subseteq \mathrm{FO}(+,\times,Q^{\mathrm{un}}_{\mathrm{Grp}})$.

Proof. TC^0 is captured by first-order logic with bit and majority quantifiers [BIS90]. Majority, however, is a context-free language. ❑

Hence arithmetic is expressive and will be difficult to defeat. The next lemma is not surprising, but it documents the provable expressiveness of arithmetic. Recall that $\mathrm{CFL} = Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}$ (Theorem 4.5).

Lemma 4.13. $\mathrm{CFL} \subsetneq Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}(+,\times)$.

Proof. The language of all words whose length is a power of two is in $\mathrm{FO}(+,\times)$, hence in the difference of the two classes. ❑

The remainder of this subsection is devoted to documenting a more complicated setting in which arithmetic provably adds expressiveness. We want to show that $\mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}}) \subsetneq \mathrm{FO}(+,\times,Q^{\mathrm{un}}_{\mathrm{Grp}})$, i.e., that even when unary groupoidal quantifiers can be nested arbitrarily, arithmetic adds strength.

For this, we define, for 0-1-strings u, w of equal length the operations $\overline{u}$, $u \wedge w$ and $u \vee w$ which denote the bitwise complementation of u, the bitwise AND of u and w and the bitwise OR of u and w. We say that a string w is (l,m)-*bounded* if it is in $u_1^* \cdots u_l^*$, for some strings u_i with $|u_i| \leq m$, for every i.

We are going to make use of the following lemma.

Lemma 4.14. *Let u be an (l,m)-bounded 0-1-string and w an (l',m')-bounded 0-1-string, for some $l, m, l', m' \geq 1$, and $|u| = |w|$. Then the following hold.*

(a) $\overline{u}$ is (l, m)-bounded.

(b) $u \wedge w$ and $u \vee w$ are $(5(l + l'), mm')$-bounded.

Proof. (a) is trivial. We show (b) only for $u \wedge w$, the argument for $u \vee w$ being completely analogous.

We show the statement by induction on $l + l'$. The induction starts with the case $l = l' = 1$.

In this case, $u = u_1^i$ and $w = w_1^j$, for some i, j, u_1, w_1, with $|u_1| \leq m$ and $|w_1| \leq m'$.

Let $u_1 \diamond w_1$ denote the string $u_1^{|w_1|} \wedge w_1^{|u_1|}$ of length $|u_1||w_1| \leq mm'$. Further let d and r be chosen such that $|u| = d|u_1||w_1| + r$ and $r < mm'$. Then $u \wedge w = (u_1 \diamond w_1)^d v$ for some v with $|v| = r$. Hence $u \wedge w$ is $(2, mm')$-bounded.

Now let $l + l' > 2$. W.l.o.g. we can assume that $u = u_1^{i_1} u_2^{i_2} u'$ and $w = w_1^j w'$ where $|u_1|, |u_2| \leq m$, $|w_1| \leq m'$, u' is $(l-2, m)$-bounded, w' is $(l'-1, m')$-bounded and $|w_1^j| \geq |u_1^{i_1}|$.

Let $0 \leq r < m'$ be such that $|u_1^{i_1}| + r$ is a multiple of $|w_1|$. Let $u_2^{\leftarrow r}$ be the word u_2 rotated r positions to the left. It should be clear that, from position $|u_1^{i_1}| + r$ in $u \wedge w$ onwards, the word $(u_2^{\leftarrow r} \diamond w_1)$ is repeated, as long as the $u_2^{i_2}$ portion of u and the w_1^j portion of w keep "overlapping". We distinguish two cases.

Case 1: $|w_1^j| \leq |u_1^{i_1}| + |u_2^{i_2}|$, i.e. the "overlap" with $u_2^{i_2}$ runs out within w_1^j.

<table>
<tr><td>$u =$</td><td colspan="2">$u_1^{i_1}$</td><td colspan="2">$u_2^{i_2}$</td><td colspan="2">u'</td></tr>
<tr><td>$w =$</td><td colspan="3">w_1^j</td><td colspan="3">w'</td></tr>
</table>

There are i_2', i_2'', u_3, u_4 with $|u_3|, |u_4| < m$, such that

<table>
<tr><td>$u =$</td><td colspan="2">$u_1^{i_1}$</td><td>$u_2^{i_2'} u_3$</td><td>$u_4 u_2^{i_2''}$</td><td colspan="2">u'</td></tr>
<tr><td>$w =$</td><td colspan="3">w_1^j</td><td colspan="3">w'</td></tr>
</table>

It is not hard to see that we can write $(u_1^{i_1} u_2^{i_2'} u_3) \wedge w_1^j$ as

$$(u_1 \diamond w_1)^{k_1} v_1 v_2 (u_2^{\leftarrow r} \diamond w_1)^{k_2} v_3,$$

for some v_2 of length r, some k_1, k_2, and some v_1, v_3 of length at most mm'. As $u_4 u_2^{i_2''} u'$ is (l, m)-bounded and w' is $(l'-1, m')$-bounded it follows by induction that $u_4 u_2^{i_2''} u' \wedge w'$ is $(5(l + l' - 1), mm')$-bounded. Altogether, $u \wedge w$ is $(5(l + l'), mm')$-bounded, as required.

Case 2: $|w_1^j| > |u_1^{i_1}| + |u_2^{i_2}|$, i.e., $u_2^{i_2}$ runs out first.

<table>
<tr><td>$u =$</td><td colspan="2">$u_1^{i_1}$</td><td colspan="2">$u_2^{i_2}$</td><td colspan="2">u'</td></tr>
<tr><td>$w =$</td><td colspan="5">w_1^j</td><td>w'</td></tr>
</table>

Hence, there are j', j'' and w_2, w_3 with $|w_2|, |w_3| < m'$ such that

<table>
<tr><td>$u =$</td><td>$u_1^{i_1}$</td><td>$u_2^{i_2}$</td><td colspan="2">u'</td></tr>
<tr><td>$w =$</td><td colspan="2">$w_1^{j'} w_2$</td><td>$w_3 w_1^{j''}$</td><td>w'</td></tr>
</table>

Now, $u_1^{i_1} u_2^{i_2} \wedge (w_1^{j'} w_2)$ can be written as

$$(u_1 \diamond w_1)^{k_1} v_1 v_2 (u_2^{\leftarrow r} \diamond w_1)^{k_2} v_3,$$

where $|v_2| = r$ and $|v_1|, |v_3| < mm'$, hence this string is $(5, mm')$-bounded. Again, by induction, it follows that the remaining part of $u \wedge w$ is $(5(l + l' - 1), mm')$-bounded, which implies the statement of the lemma. ❑

Let Σ be a fixed alphabet, and let σ denote the corresponding signature. Let φ be a FO(+)-formula over σ with free variables x and $\overline{y} = y_1, \ldots, y_k$. For every string $w \in \Sigma^*$ and k-tuple $\overline{q}$ of positions in w, we write $t_\varphi^{\overline{q}}(w)$ for the 0-1 string $v = v_1, \ldots v_{|w|}$ with $v_i = 1$ iff $\langle w, i, \overline{q} \rangle \models \varphi$.

Lemma 4.15. *Let $\Sigma = \{0\}$ and $\sigma_0 = \{P_0\}$. Let φ be a* FO(+)*-formula over σ_0 with free parameters x and $\overline{y} = y_1, \ldots, y_k$. Then there are l and m such that for every n and $y_1, \ldots, y_k$ it holds that $t_\varphi^{\overline{q}}(0^n)$ is (l, m)-bounded.*

Proof. Let φ' be the FO(+)-formula over the empty signature which results from φ by replacing every subformula $P_0(t)$ by true, introducing a new free variable, n, and restricting all quantifiers relative to n. I. e., subformulas $\exists z \theta$ are replaced by $\exists z (z < n) \wedge \theta$ and $\forall z \theta$ is replaced by $\forall z (z < n) \rightarrow \theta$. Then we get

$$\langle 0^n, x, \overline{q} \rangle \models \varphi \iff \langle \mathbb{N}, n, x, \overline{q} \rangle \models \varphi',$$

where $\mathbb{N}$ denotes the natural numbers. Using Presburger Quantifier Elimination (see [BJ89, pp. 220ff] or [Smo87, pp. 320ff]) we can transform φ' into an equivalent quantifier-free formula ψ which may additionally use the constants 0 and 1 and binary predicates $\cdot \equiv \cdot \pmod{c}$, for some constants c. The atomic formulas of ψ are of one of the following forms.

- $ax + bn + a_1 y_1 + \cdots a_k y_k = c$,
- $ax + bn + a_1 y_1 + \cdots a_k y_k < c$,
- $ax + bn + a_1 y_1 + \cdots a_k y_k > c$,
- $ax + bn + a_1 y_1 + \cdots a_k y_k \equiv c \pmod{d}$,

for some constants a, b, c, d, a_i. For every fixed n and $q_1, \ldots, q_k$, the first formula defines (by substituting each y_i by q_i), via the above equivalence, a $(3,1)$-bounded string in 0^*10^*, the second and third formula define a $(2,1)$-bounded string in 1^*0^* and 0^*1^* respectively, and the last formula defines a $(2, d)$-bounded string in $0^*(10^{d-1})^*$. As ψ is fixed, by inductively applying Lemma 4.14 we get constants l and m, such that, for every $n, \overline{q}$, $t_\varphi^{\overline{q}}(0^n)$ is (l, m)-bounded. ❑

Theorem 4.16. $\mathrm{FO}(+,\times,Q^{\mathrm{un}}_{\mathrm{Grp}})$ *is not contained in* $\mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}})$.

Proof. We consider the language $\{\, 0^{n^2} \mid n \in \mathbb{N} \,\}$, which is even expressible in $\mathrm{FO}(+,\times)$ and show that it is not in $\mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}})$.

In order to do so, we show that, for every unary language L in $\mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}})$, the set $\{\, i \mid 0^i \in L \,\}$ is semi-linear (i.e., the finite union of some arithmetic progressions). It is enough to show that, over a one-letter alphabet, every formula of the kind $Q_B x\varphi$ with CFL B and first-order φ (with addition) can be replaced by a first-order formula with addition.

Hence, let $\psi = Q_B x\varphi$, for some first-order ϕ (with addition) and CFL B. Let, besides x, $\overline{y} = y_1, \dots, y_k$ be the free variables of φ.

By Lemma 4.15, there exist l and m such that, for every n and $\overline{q}$, $t^{\overline{q}}_{\varphi}(0^n)$ is (l,m)-bounded. Let $u_1, \dots, u_p$ be an enumeration of all 0-1 strings of length at most m. Let L' denote the (regular) language, defined by $(u_1^* \cdots u_p^*)^l$. It follows that $t^{\overline{q}}_{\varphi}(0^n)$ is in L', hence it can be written as $u_1^{i_{11}} \cdots u_p^{i_{1p}} u_1^{i_{21}} \cdots u_p^{i_{2p}} \cdots u_p^{i_{lp}}$ (where, for each $j = 1, \dots, l$, all but one of the $i_{j1}, \dots, i_{jp}$ are 0). For a word $w \in L'$ we write $I(w)$ for the set of tuples $(i_{11}, \dots, i_{lp})$ with $u_{11}^{i_{11}} \cdots u_{lp}^{i_{lp}} = w$. We show in the following that $I_B := \bigcup_{w \in B \cap L'} I(w)$ is a semi-linear set.

Let $\Gamma = (a_{11}, \dots, a_{1p}, \dots, a_{lp})$ be a new (lp-letter) alphabet and let h be the homomorphism defined by $h(a_{ij}) = u_i$. Let τ denote the Parikh mapping for strings $a_{11}^* \cdots a_{lp}^*$. Then we have

$$I_B = \tau(h^{-1}(B \cap L') \cap a_{11}^* \cdots a_{lp}^*),$$

which is semi-linear by Parikh's theorem [Har78, Sect. 6.9].

Hence, ψ is equivalent to a $\mathrm{FO}(+)$ formula [Har78, p. 231]. By induction, we get that every $\mathrm{FO}(+,Q^{\mathrm{un}}_{\mathrm{Grp}})$, hence also every $\mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}})$ formula, over a one-letter alphabet is equivalent to a $\mathrm{FO}(+)$ formula. Hence $\{\, 0^{n^2} \mid n \in \mathbb{N} \,\}$ is not in $\mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}})$. ❑

It is interesting to see that the proof makes use of quantifier elimination twice, first to get the bounded strings, and second to show that $\{\, 0^{n^2} \mid n \in \mathbb{N} \,\}$ is not in $\mathrm{FO}(+)$.

As a particular case we can now solve an open question of [BIS90], addressing the power of different arity for majority quantifiers.

Corollary 4.17. *Majority of pairs can not be expressed in first-order logic with unary majority quantifiers.*

Proof. In Theorem 4.10 it was observed that the unary majority quantifier can be simulated in $\mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}})$. On the other hand, in [BIS90] it is shown that majority of pairs is sufficient to simulate the BIT predicate. But as $\mathrm{FO}(+,\times,Q^{\mathrm{un}}_{\mathrm{Grp}}) = \mathrm{FO}(\mathrm{BIT},Q^{\mathrm{un}}_{\mathrm{Grp}})$ is not contained in $\mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}})$, the BIT predicate and hence the majority of pairs is not definable in $\mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}})$, hence it cannot be simulated by unary majority quantifiers. ❑

In the same way, this time relying on Theorem 4.11, we obtain:

Corollary 4.18. *Multiplication is not definable in* $\mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}})$.

5 Conclusion

Fig. 1 depicts the first-order groupoidal-quantifier-based classes studied in this article. Together with the new characterization of FO-translations by means of aperiodic finite transducers, the relationships shown on Fig. 1 summarize our contribution.

A number of open questions are apparent from Figure 1. Clearly, it would be nice to separate the $\mathrm{FO}(+,\times)$-based classes, in particular $\mathrm{FO}(+,\times,Q^{\mathrm{un}}_{\mathrm{Grp}})$ from $\mathrm{FO}(+,\times,Q_{\mathrm{Grp}})$, but this is a daunting task. A sensible approach then is to begin with $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}(+,\times)$. How does this compare with TC^0 for example? Can we at least separate $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}(+,\times)$ from LOGCFL? We know that $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}(+,\times) \not\subseteq \mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}})$; a witness for this is the set $\left\{ 0^{n^2} \;\middle|\; n \in \mathbb{N} \right\}$, cf. the proof of Theorem 4.16.

Other natural questions prompted by our separation results concern extensions and refinements to Figure 1. For example, in the world with arithmetic, which specific groupoids G are powerful enough to express LOGCFL, and which are not? In the world without arithmetic, given the aperiodic transducer characterization of FO-translations, can we prove $\mathrm{REG} \not\subseteq Q^{\mathrm{un}}_{G}\mathrm{FO}$ as easily as Lemma 4.8 implies $\mathrm{CFL} \not\subseteq Q^{\mathrm{un}}_{G}\mathrm{FO}$?

When we relate Fig. 1 to the world of monoids, the following parallels and differences come to mind: Similar to our case, the circuit class NC^1 equals $\mathrm{FO}(+,\times,Q_{\mathrm{Mon}}) = Q_{\mathrm{Mon}}\mathrm{FO}(+,\times)$. However, arithmetic is provably needed in this characterization. Already in [BIS90], the equality $\mathrm{REG} = Q^{\mathrm{un}}_{\mathrm{Mon}}\mathrm{FO}$ is given. Much different from corresponding results given in this article about CFL, the following can even be shown:

Theorem 5.1. $\mathrm{REG} = \mathrm{FO}(Q_{\mathrm{Mon}})$.

Proof. (Sketch.) As a first step, let $A \in Q_{\mathrm{Mon}}\mathrm{FO}$ be defined by the formula

$$Q_B\overline{x}\,[\phi_1(\overline{x}),\ldots,\phi_{s-1}(\overline{x})]\,.$$

Let us suppose for simplicity that Q_B binds two variables. The transformation $[\phi_1(\overline{x}),\ldots,\phi_{s-1}(\overline{x})]$ maps an input word x to a matrix M_x, and the question is if the string obtained by concatenating the rows of M_x is an element of B.

We construct a monadic second-order formula that defines A. Let M be a finite automaton that accepts B, with transition function δ and state set Q. Let $z_1,\ldots,z_{|x|}$ be the rows of M_x. Let $f_1,\ldots,f_{|x|}\colon Q \to Q$ be the transformations in M induced by reading the words $z_1,\ldots,z_{|x|}$. A second-order existential quantifier is used to "guess" these transformations. That the guesses are correct reduces to a regular word problem, hence can be expressed by a monadic second-order formula. Finally, it remains to check that the product of $f_1, \ldots, f_{|x|}$ is a transformation that maps the initial state of M to a final state. Again, this is a regular word problem and can be expressed by a suitable formula.

The generalization to higher dimensions is straightforward. For example in three dimensions the transformation $[\phi_1(\overline{x}), \ldots, \phi_{s-1}(\overline{x})]$ maps an input word to a cube of size $|x|^3$. Here we first guess the transformations for every plane in the cube; to verify that these guesses are correct we need another existential quantifier to treat each plane in the same way as the matrix M_x above. This shows $Q_{\mathrm{Mon}}\mathrm{FO} \subseteq \mathrm{REG}$. The claim of the theorem now follows by an easy induction. ❑

Additionally, we remark that—analogous to our groupoidal case—the classes $Q^{\mathrm{un}}_{\mathrm{Mon}}\mathrm{FO}(+,\times)$ and $\mathrm{FO}(+,\times,Q^{\mathrm{un}}_{\mathrm{Mon}})$ are (trivially) strict superclasses of REG and subclasses of NC^1, but nothing more is known.

Given the analogies and differences between the world of monoids and the world of groupoids, it is clearly an intriguing question if we can hope for an algebraic theory of groupoids to explain the detailed structure of CFL (and SAC^1), much in the way that an elaborate theory of monoids is used in the extensive first-order parameterization of REG (and NC^1)?

But perhaps the most fundamental (and hopefully tractable) question arising from our work is not apparent from Figure 1. It concerns the Boolean closure of the context-free languages. We have trivially used BC(CFL) (in fact $\mathrm{BC}^+(\mathrm{CFL})$ sufficed) to witness the separation between $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}$ and $\mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}})$. But what is BC(CFL) exactly, and what techniques are available to prove that a language is not in BC(CFL)? It is easy to prove that any non-regular language over a unary alphabet does not belong to BC(CFL), and a natural infinite hierarchy within $\mathrm{BC}^+(\mathrm{CFL})$ is known [LW73], but the full question seems to have fallen into the cracks. We have several good candidates for membership in $\mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}}) \setminus \mathrm{BC}(\mathrm{CFL})$, but so far have been unable to prove these two classes different.

Finally, ever since the regular languages in AC^0 and in ACC^0 were characterized (the latter modulo a natural conjecture [BCST92]), one might have wondered about a similar characterization for the context-free languages in these classes, and in NC^1. A unified treatment of LOGCFL subclasses under the banner of first-order logic might constitute a useful step towards being able to answer these questions. Since circuit-based complexity classes are closed under Boolean operations however, a better understanding of the interaction between the complement operation and groupoidal quantifiers is required. This once again seems to highlight the importance of understanding BC(CFL).

Acknowledgments. We thank Dave Barrington, Gerhard Buntrock, Volker Diekert, Klaus-Jörn Lange, Ken Regan, Heinz Schmitz, Denis Thérien, Wolfgang Thomas, Klaus Wagner, and Detlef Wotschke for useful discussions at one stage or another in the course of this work. We further thank an anonymous referee for a very careful reading of our manuscript and for many judicious comments regarding ways to make the article and results more accessible to a complexity theory audience

References

[AAI+97] M. Agrawal, E. Allender, R. Impagliazzo, R. Pitassi, and S. Rudich. Reducing the complexity of reductions. In *Proceedings 29th Symposium on Theory of Computing*, pages 730–738. ACM Press, 1997.

[ABB97] J.-M. Autebert, J. Berstel, and L. Boasson. Context-free languages and pushdown automata. In R. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume I, chapter 3, pages 111–174. Springer Verlag, Berlin Heidelberg, 1997.

[Ajt83] M. Ajtai. Σ_1^1 formulae on finite structures. *Annals of Pure and Applied Logic*, 24:1–48, 1983.

[Bar89] D. A. Mix Barrington. Bounded-width polynomial size branching programs recognize exactly those languages in NC^1. *Journal of Computer and System Sciences*, 38:150–164, 1989.

[BCST92] D. A. Mix Barrington, K. Compton, H. Straubing, and D. Thérien. Regular languages in NC^1. *Journal of Computer and System Sciences*, 44:478–499, 1992.

[BIS90] D. A. Mix Barrington, N. Immerman, and H. Straubing. On uniformity within NC^1. *Journal of Computer and System Sciences*, 41:274–306, 1990.

[BJ89] G. S. Boolos and R. C. Jeffrey. *Computability and Logic*. Cambridge University Press, 1989.

[BLM93] F. Bédard, F. Lemieux, and P. McKenzie. Extensions to Barrington's M-program model. *Theoretical Computer Science*, 107:31–61, 1993.

[BT88] D. A. Mix Barrington and D. Thérien. Finite monoids and the fine structure of NC^1. *Journal of the Association of Computing Machinery*, 35:941–952, 1988.

[BV98] H.-J. Burtschick and H. Vollmer. Lindström quantifiers and leaf language definability. *International Journal of Foundations of Computer Science*, 9:277–294, 1998.

[Coo71] S. A. Cook. Characterizations of pushdown machines in terms of time-bounded computers. *Journal of the Association for Computing Machinery*, 18:4–18, 1971.

[DDLW98] A. Dawar, K. Doests, S. Lindell, and S. Weinstein. Elementary properties of the finite ranks. *Mathematical Logic Quarterly*, 44:349–353, 1998.

[EF95] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Perspectives in Mathematical Logic. Springer Verlag, Berlin Heidelberg, 1995.

[EFT94] H.-D. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical Logic*. Springer Verlag, Berlin Heidelberg, 2nd edition, 1994.

[FSS84] M. Furst, J. B. Saxe, and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17:13–27, 1984.

[Gre73] S. Greibach. The hardest context-free language. *SIAM Journal on Computing*, 2:304–310, 1973.

[Har78] M. A. Harrison. *Introduction for Formal Language Theory*. Addison-Wesley, Reading, MA, 1978.

[Imm89] N. Immerman. Expressibility and parallel complexity. *SIAM Journal on Computing*, 18:625–638, 1989.

[Imm99] N. Immerman. *Descriptive Complexity*. Graduate Texts in Computer Science. Springer Verlag, New York, 1999.

[Lin66] P. Lindström. First order predicate logic with generalized quantifiers. *Theoria*, 32:186–195, 1966.

[Lin95] S. Lindell. Unary counting does not suffice. Manuscript, 1995.

[LST94] C. Lautemann, T. Schwentick, and D. Thérien. Logics for context-free languages. In L. Pacholski and J. Tiuryn, editors, *8th Computer Science Logic, Selected Papers*, volume 933 of *Lecture Notes in Computer Science*, pages 205–216. Springer Verlag, 1994.

[LW73] L. Liu and P. Weiner. An infinite hierarchy of intersections of context-free languages. *Mathematical Systems Theory*, 7:185–192, 1973.

[MP71] R. McNaughton and S. Papert. *Counter-Free Automata*. MIT Press, 1971.

[MS97] A. Mateescu and A. Salomaa. Aspects of classical language theory. In R. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume I, chapter 4. Springer Verlag, Berlin Heidelberg, 1997.

[PMT91] P. Péladeau P. McKenzie and D. Thérien. NC^1: The automata-theoretic viewpoint. *Computational Complexity*, 1:330–359, 1991.

[Sch65] M. P. Schützenberger. On finite monoids having only trivial subgroups. *Information & Control*, 8:190–194, 1965.

[Smo87] R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings 19th Symposium on Theory of Computing*, pages 77–82. ACM Press, 1987.

[Ste85] J. Stern. Complexity of some problems from the theory of automata. *Information & Computation*, 66:163–176, 1985.

[Str94] H. Straubing. *Finite Automata, Formal Logic, and Circuit Complexity.* Birkhäuser, Boston, 1994.

[Sud78] I. H. Sudborough. On the tape complexity of deterministic context-free languages. *Journal of the Association for Computing Machinery*, 25:405–414, 1978.

[Ven91] H. Venkateswaran. Properties that characterize LOGCFL. *Journal of Computer and System Sciences*, 43:380–404, 1991.

[Vol99] H. Vollmer. *Introduction to Circuit Complexity – A Uniform Approach.* Texts in Theoretical Computer Science. Springer Verlag, Berlin Heidelberg, 1999.

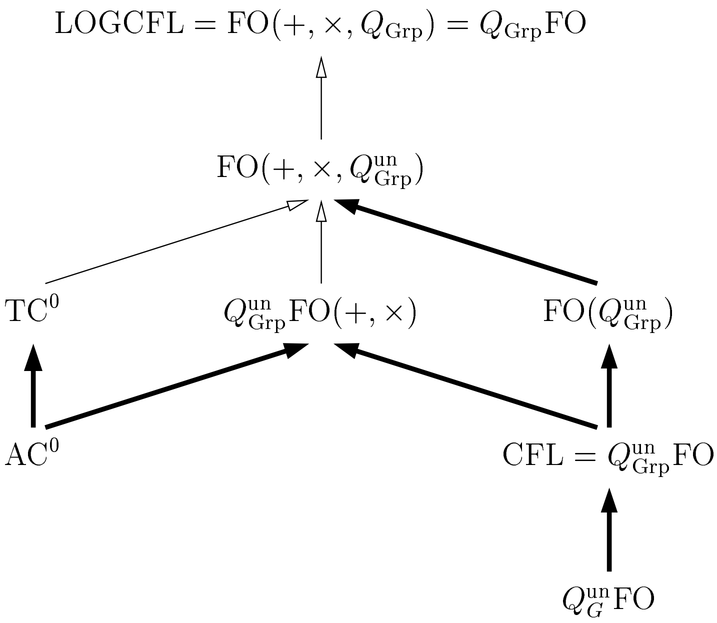

Figure 1: The new landscape. Here G stands for any fixed groupoid, and a thick line indicates strict inclusion.