The Descriptive Complexity Approach to LOGCFL

Clemens Lautemann[1], Pierre McKenzie[2*], Thomas Schwentick[1], and Heribert Vollmer[3]

[1] Institut für Informatik, Johannes-Gutenberg-Universität Mainz, 55099 Mainz, Germany.
[2] Informatique et recherche opérationnelle, Université de Montréal, C.P. 6128, Succ. Centre-Ville, Montréal (Québec), H3C 3J7 Canada.
[3] Theoretische Informatik, Universität Würzburg, Am Exerzierplatz 3, 97072 Würzburg, Germany.

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. Our work extends the elaborate theory relating *monoidal* quantifiers to NC^1 and its subclasses. 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 BIT-free interpretations. We then prove that FO with unary groupoidal quantifiers is strictly more expressive with the BIT predicate 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.

1 Introduction

In *Finite Automata, Formal Logic, and Circuit Complexity* [17], 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 from 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 translation provides new insights, makes tools from one field available

* Research performed while on leave at the Universität Tübingen. Supported by the (German) DFG, the (Canadian) NSERC and the (Québec) FCAR.

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.

In this way, although, for example, the unresolved strict containment in NC^1 of the class ACC^0, defined from bounded-depth polynomial-size unbounded fan-in circuits over $\{\mathrm{AND}, \mathrm{OR}, \mathrm{MOD}\}$, remains a barrier since the work of Smolensky [15], significant progress was made in (1) understanding the power of the BIT predicate and the related circuit uniformity issues [3], (2) describing the regular languages within subclasses of NC^1 [2, 13], and (3) 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 [17, p. 169, Conjecture IX.3.4].

Barrington, Immerman and Straubing [3] introduced the notion of a *monoidal* quantifier and noted that, for any non-solvable group G, the class NC^1 can be described using first-order logic augmented with a monoidal quantifier for G. Loosely speaking, such a quantifier provides a constrained "oracle call" to the word problem for G (defined essentially as the problem of computing the product of a sequence of elements of G).

Bédard, Lemieux and McKenzie [4] later noted 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 [6, 18]. 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 [4, Lemma 3.1]).

It followed that LOGCFL, a well-studied class which contains nondeterministic logarithmic space [18] 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 [10] for context-free languages.

In this paper, we take up this first-order characterization of LOGCFL, and initiate an investigation of LOGCFL from the viewpoint of descriptive complexity. The rationale for this study, which encompasses the study 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 BIT predicate. Feature (3) was the focus of an important part of the work by Barrington, Immerman and Straubing [3] 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 [3, 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 the BIT predicate, we are able to determine the following relationships:

- 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 [3]: We show that first-order logic with the majority-of-pairs quantifier is strictly more expressive than first-order logic with majority of individuals, see Corollary 18.
- 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,
- 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 [7], see also [1], yields a language which is LOGCFL-complete under BIT-free quantifier-free interpretations.

When the BIT predicate is present, first-order with non-unary groupoidal quantifiers of course still describes LOGCFL. In the setting of monoidal quantifiers [3], FO with BIT 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: the BIT predicate allows capturing classes (for example $\mathrm{FO}_{\mathrm{bit}}(Q^{\mathrm{un}}_{\mathrm{Grp}})$, verifying $\mathrm{TC}^0 \subseteq \mathrm{FO}_{\mathrm{bit}}(Q^{\mathrm{un}}_{\mathrm{Grp}}) \subseteq \mathrm{LOGCFL}$), which only a major breakthrough would seem to allow separating from each other. We are able to attest to the strength of the BIT predicate in the setting of unary quantifiers, proving that:

- $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO} \subsetneq Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}_{\mathrm{bit}}$, i. e. (trivially) some non-context-free languages are expressible using BIT and a single unary groupoidal quantifier,
- $\mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}}) \subsetneq \mathrm{FO}_{\mathrm{bit}}(Q^{\mathrm{un}}_{\mathrm{Grp}})$, i. e. (more interestingly) BIT adds 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 [14], but, to the best of our knowledge, always in a deterministic context. Here we define a NFA A to be aperiodic 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.

Due to lack of space most proofs are omitted in this abstract. A full paper including complete proofs of all our claims can be obtained as ECCC Report 98-59 from http://www.eccc.uni-trier.de.

2 Preliminaries

2.1 Complexity Theory

REG and CFL refer to the regular and to the ϵ-free context-free languages respectively. The CFL results in this paper could be adapted to treat the empty string ϵ in standard ways. We will make scant reference to the inclusion chain $\mathrm{AC}^0 \subsetneq \mathrm{ACC}^0 \subseteq \mathrm{TC}^0 \subseteq \mathrm{NC}^1 \subseteq \mathrm{NL} \subseteq \mathrm{LOGCFL} = \mathrm{SAC}^1 \subseteq \mathrm{P}$. We refer the reader to [17] for definitions of these classes.

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 $\langle P_{a_1}, \dots, P_{a_s}\rangle$, 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 $(a_1, \dots, a_s)$, and first-order variables range over positions within such a word, i. e. from 1 to the word length n. For technical reasons that will become apparent shortly, we assume here, as in the rest of the paper, a linear order on each alphabet and we write alphabets as sequences of symbols to indicate that order.

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. We will occasionally use the binary predicate $\mathrm{BIT}(x, y)$, defined to be true iff the xth bit in the binary representation of y is 1. We 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.

Definition 1. *Lindström quantifier.* Consider a language L over an alphabet $\Sigma = (a_1, a_2, \dots, a_s)$, $s > 1$. 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 $\{1, 2, \dots, n\}^k$, and we write $X_1, X_2, \dots, X_{n^k}$ for the sequence of potential values taken on by $\overline{x}$. The k-ary groupoidal quantifier Q_L binding $\overline{x}$ takes a meaning if $s-1$ formulas, each having as free variables the variables in $\overline{x}$ (and possibly others), are available. Let $\phi_1(\overline{x})$, $\phi_2(\overline{x})$, ..., $\phi_{s-1}(\overline{x})$ be these $s-1$ formulas. Then $Q_L\overline{x}\big[\phi_1(\overline{x}), \phi_2(\overline{x}), \dots, \phi_{s-1}(\overline{x})\big]$ holds on a string $w = w_1 \cdots w_n$ (where $w_i \in \Sigma$ for all i), iff the word of length n^k whose ith letter, $1 \le i \le n^k$, is

$$\begin{cases} a_1 \text{ if } w \models \phi_1(X_i), \\ a_2 \text{ if } w \models \neg\phi_1(X_i) \wedge \phi_2(X_i), \\ \quad \cdots \\ a_s \text{ if } w \models \neg\phi_1(X_i) \wedge \neg\phi_2(X_i) \wedge \ldots \wedge \neg\phi_{s-1}(X_i), \end{cases}$$

belongs to L. Thus the formulas $[\phi_1(\overline{x}), \phi_2(\overline{x}), \dots, \phi_{s-1}(\overline{x})]$ fix a function mapping an input word/structure w of length n to a word of length n^k. This function is called the *reduction* or *transformation* defined by $[\phi_1(\overline{x}), \phi_2(\overline{x}), \dots, \phi_{s-1}(\overline{x})]$. In case we deal with the binary alphabet ($s = 2$) we omit the braces and write $Q_L \overline{x} \phi(\overline{x})$ for short.

Definition 2. A *groupoidal quantifier* is a Lindström quantifier Q_L where L is a context-free language.

The Lindström quantifiers of Definition 1 are more precisely what has been refered to as "Lindström quantifiers on strings" [5]. The original more general definition [10] uses transformations to arbitrary structures, not necessarily of string signature. However, in the context of this paper reductions to CFLs play a role of utmost importance, and hence the above definition seems to be the most natural. The terminology *groupoidal quantifier* stems from the fact that any context-free language is a word problem over some groupoid [4, Lemma 3.1], and vice-versa every word problem of a finite groupoid is context-free. Thus a Lindström quantifier on strings defined by a context-free language is nothing else than a Lindström quantifier (in the classical sense) defined by a structure that is a finite groupoid multiplication table.

Barrington, Immerman, and Straubing, defining *monoidal quantifiers* in [3], in fact proceed along the same avenue: they first show how monoid word problems can be seen as languages, and then define generalized quantifiers given by such languages (see [3, pp. 284f.]).

2.3 Unary Quantifiers and Homomorphisms

We will encounter unary groupoidal quantifiers repeatedly. Here we show how these relate to standard formal language operations. In a different context, a result very similar to the next theorem is known as *Nivat's Theorem* [11, Theorem 3.8, p. 207].

Theorem 3. *Let B be an arbitrary language, and let A be describable in $Q_B^{\mathrm{un}}\mathrm{FO}$, that is, by a first order formula preceded by one unary Lindström quantifier (i. e. binding exactly one variable). Then there are length-preserving homomorphisms g, h and a regular language D such that $A = h(D \cap g^{-1}(B))$.*

2.4 Groupoid-Based Language Classes

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

Definition 4. $Q_G\mathrm{FO}$ is the set of languages describable by applying a single groupoidal quantifier Q_L to an appropriate tuple of FO formulas, where $L = \mathcal{W}(S, G)$ for some $S \subseteq G$.
$Q_{\mathrm{Grp}}\mathrm{FO}$ is the union, over all finite groupoids G, of $Q_G\mathrm{FO}$.
$\mathrm{FO}(Q_G)$ and $\mathrm{FO}(Q_{\mathrm{Grp}})$ are defined analogously, but allowing groupoidal quantifiers to be used as any other quantifier would (i. e. allowing arbitrary nesting).

$Q_G^{\mathrm{un}}\mathrm{FO}$ and $\mathrm{FO}_{\mathrm{bit}}(Q_{\mathrm{Grp}}^{\mathrm{un}})$, etc, are defined analogously, but possibly allowing the BIT predicate (signaled by subscripting FO with bit) and/or restricting to *unary* groupoidal quantifiers (signaled by the superscript "un").

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 [12], 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 5. 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 6. This definition of aperiodicity for a DFA is the usual one (see [16]). For a NFA, a statement obviously equivalent to Definition 5 would be that A is aperiodic iff applying the subset construction to A yields an aperiodic DFA. Hence [14] a language L is star-free iff some aperiodic (deterministic or nondeterministic) finite automaton accepts L.

Definition 7. A *finite transducer* M 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 $s_0 = q_0, s_1, \ldots, s_n$ of states, such that $s_n \in F$ and, for every i, $1 \leq i \leq n$, we have $(s_{i-1}, w_i, v_i, s_i) \in \delta$.

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 $s_1, \ldots, s_{|u|} = s'$ 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 5, M is *aperiodic* if there is an $n \in \mathbb{N}$ such that for all states q and all strings w, $\delta^*(q, w^n) = \delta^*(q, w^{n+1})$.

Theorem 8. *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.*

4 First-Order with Groupoidal Quantifiers

4.1 Non-Unary Groupoidal Quantifiers

Clearly all classes from our framework fall within LOGCFL. But even one groupoidal quantifier to the left is sufficient:

Theorem 9. *There is a fixed groupoid G such that $Q_G\mathrm{FO}_{\mathrm{bit}} = \mathrm{FO}_{\mathrm{bit}}(Q_{\mathrm{Grp}}) = \mathrm{LOGCFL}$.*

As we show next, the largest attainable class LOGCFL is even characterizable without bit, as long as we allow arbitrary arities of the quantifiers:

Theorem 10. *There is a fixed groupoid G such that $\mathrm{LOGCFL} \subseteq Q_G\mathrm{FO}$, hence $Q_{\mathrm{Grp}}\mathrm{FO} = \mathrm{FO}(Q_{\mathrm{Grp}}) = \mathrm{LOGCFL}$.*

Corollary 11. *Greibach's hardest context-free language with a neutral symbol is complete for* LOGCFL *under quantifier-free interpretations without BIT.*

4.2 Unary Groupoidal Quantifiers without BIT

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 BIT predicate, captures all of LOGCFL. Here we examine the case of unary quantifiers. Let us first turn to the case of unary groupoidal quantifiers without BIT.

Theorem 12. $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO} = \mathrm{CFL}$.

Proof. The direction from right to left is obvious. The converse direction is proved by appealing to Theorem 3 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:

Corollary 13.
$$\begin{aligned} Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO} = \mathrm{CFL} &\subsetneq \mathrm{BC}^+(Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}) = \mathrm{BC}^+(\mathrm{CFL}) \\ &\subsetneq \mathrm{BC}(Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}) = \mathrm{BC}(\mathrm{CFL}) \\ &\subseteq \mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}}). \end{aligned}$$

Can we refine Theorem 12 and find a universal finite groupoid G which captures all the context-free languages as $Q^{\mathrm{un}}_G\mathrm{FO}$? Intuition from the world of monoids [3, p. 303] suggests that the answer is no. Proving that this is indeed the case is the content of Theorem 15 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 which 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 14. *No k-pushdown-limited PDA accepts D_t when $t \geq (k+1)^k + 1$.*

Theorem 15. *Any finite groupoid G verifies $Q_G^{\mathrm{un}}\mathrm{FO} \subsetneq \mathrm{CFL}$.*

Proof. Suppose to the contrary that G is a finite groupoid such that $Q_G^{\mathrm{un}}\mathrm{FO} = \mathrm{CFL}$. 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 $\mathcal{W}(\cdot, G)$) can take care of answering each "oracle question" resulting from such a FO-translation. By Theorem 8, 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 $\mathcal{W}(\cdot, G)$ 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 14 when $t = (k+1)^k + 1$.

In the next subsection we will see that the BIT-predicate provably adds expressive power to the logic $Q_{\mathrm{Grp}}^{\mathrm{un}}\mathrm{FO}$. Since it is known that BIT can be expressed either by plus and times [9] (cf., [8]) or by the majority of pairs quantifier [3], the following two simple observations about the power of $Q_{\mathrm{Grp}}^{\mathrm{un}}\mathrm{FO}$ are of particular interest.

Theorem 16. *1. The majority quantifier is definable in $Q_{\mathrm{Grp}}^{\mathrm{un}}\mathrm{FO}$.*
2. Addition is definable in $Q_{\mathrm{Grp}}^{\mathrm{un}}\mathrm{FO}$.

4.3 Unary Groupoidal Quantifiers with BIT

What are $Q_{\mathrm{Grp}}^{\mathrm{un}}\mathrm{FO}_{\mathrm{bit}}$ and $\mathrm{FO}_{\mathrm{bit}}(Q_{\mathrm{Grp}}^{\mathrm{un}})$? It seems plausible that $Q_{\mathrm{Grp}}^{\mathrm{un}}\mathrm{FO}_{\mathrm{bit}} \subsetneq \mathrm{FO}_{\mathrm{bit}}(Q_{\mathrm{Grp}}^{\mathrm{un}}) \subsetneq \mathrm{LOGCFL}$, but we are unable to prove these separations. Since TC^0 is captured by first-order logic with bit and majority quantifiers (definable by a context-free language), we conclude $\mathrm{TC}^0 \subseteq \mathrm{FO}_{\mathrm{bit}}(Q_{\mathrm{Grp}}^{\mathrm{un}})$, hence proving the latter separation would prove $\mathrm{TC}^0 \neq \mathrm{LOGCFL}$, settling a major open question in complexity theory.

Easily $\mathrm{CFL} = Q_{\mathrm{Grp}}^{\mathrm{un}}\mathrm{FO} \subsetneq Q_{\mathrm{Grp}}^{\mathrm{un}}\mathrm{FO}_{\mathrm{bit}}$, since $\left\{ 0^{2^n} \,\middle|\, n \in \mathbb{N} \right\}$ is in the difference of the two classes. The remainder of this subsection is devoted to documenting a more complicated setting in which the BIT predicate provably adds expressiveness.

Theorem 17. $\mathrm{FO}_{\mathrm{bit}}(Q_{\mathrm{Grp}}^{\mathrm{un}})$ *is not contained in* $\mathrm{FO}(Q_{\mathrm{Grp}}^{\mathrm{un}})$.

As a particular case we can now solve an open question of [3], addressing the power of different arity for majority quantifiers.

Corollary 18. *Majority of pairs can not be expressed in first-order logic with unary majority quantifiers.*

Proof. In Theorem 16 it was observed that the unary majority quantifier can be simulated in $\mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}})$. On the other hand in [3] it is shown that majority of pairs is sufficient to simulate the BIT predicate. But as $\mathrm{FO}_{\mathrm{bit}}(Q^{\mathrm{un}}_{\mathrm{Grp}})$ 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 can not be simulated by unary majority quantifiers.

Corollary 19. *Multiplication is not definable in* $\mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}})$.

5 Conclusion

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

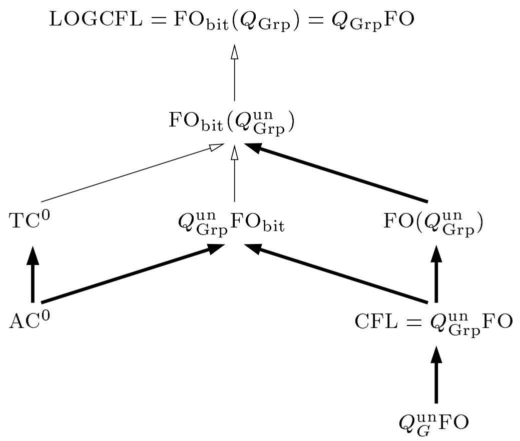

Fig. 1. The new landscape. Here G stands for any fixed groupoid, and a thick line indicates strict inclusion.

A number of open questions are apparent from Figure 1. Clearly, it would be nice to separate the $\mathrm{FO}_{\mathrm{bit}}$-based classes, in particular $\mathrm{FO}_{\mathrm{bit}}(Q^{\mathrm{un}}_{\mathrm{Grp}})$ from $\mathrm{FO}_{\mathrm{bit}}(Q_{\mathrm{Grp}})$, but this is a daunting task. A sensible approach then is to begin with $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}_{\mathrm{bit}}$. How does this compare with TC^0 for example? Can we at least separate $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}_{\mathrm{bit}}$ from LOGCFL? We know that $Q^{\mathrm{un}}_{\mathrm{Grp}}\mathrm{FO}_{\mathrm{bit}} \not\subseteq \mathrm{FO}(Q^{\mathrm{un}}_{\mathrm{Grp}})$; a witness for this is the set $\left\{ 0^{n^2} \mid n \in \mathbb{N} \right\}$, cf. the proof of Theorem 17.

An important fundamental question of course is if we can hope for an algebraic theory of groupoids to explain the detailed structure of CFL, much in the way that an elaborate theory of monoids is used in the extensive first-order parameterization of REG.

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 in the course of this work.

References

[1] 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. Springer Verlag, Berlin Heidelberg, 1997.

[2] 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.

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

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

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

[6] S. A. Cook. Characterizations of pushdown machines in terms of time-bounded computers. *Journal of the ACM*, 18:4–18, 1971.

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

[8] N. Immerman. *Descriptive Complexity*. Springer Verlag, New York, 1998.

[9] S. Lindell. manuscript, 1994. e-mail communication by Kenneth W. Regan.

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

[11] 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.

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

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

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

[15] 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.

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

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

[18] I. H. Sudborough. On the tape complexity of deterministic context-free languages. *Journal of the ACM*, 25:405–414, 1978.