The Descriptive Complexity of the Fixed-Points of Bounded Formulas

Albert Atserias*

Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
c/ Jordi Girona Salgado, 1-3, Edif. C6.
08034 Barcelona, Spain.
Email: atserias@lsi.upc.es
Telf: +34.93.401.7293
Fax: +34.93.401.7014

Abstract. We investigate the complexity of the fixed-points of bounded formulas in the context of finite set theory; that is, in the context of arbitrary classes of finite structures that are equipped with a built-in BIT predicate, or equivalently, with a built-in membership relation between hereditarily finite sets (input relations are allowed). We show that the iteration of a positive bounded formula converges in polylogarithmically many steps in the cardinality of the structure. This extends a previously known much weaker result. We obtain a number of connections with the rudimentary languages and deterministic polynomial-time. Moreover, our results provide a natural characterization of the complexity class consisting of all languages computable by bounded-depth, polynomial-size circuits, and polylogarithmic-time uniformity. As a byproduct, we see that this class coincides with LH(P), the logarithmic-time hierarchy with an oracle to deterministic polynomial-time. Finally, we discuss the connection of this result with the well-studied algorithms for integer division.

Keywords: Circuit uniformity, BIT predicate, logarithmic-time hierarchy, rudimentary languages, integer division.

* Supported by the CUR, Generalitat de Catalunya, through grant 1999FI 00532.

1 Introduction

1.1 Background

The Ordered Conjecture of Kolaitis and Vardi [22] states that least fixed-point logic LFP is strictly more expressive than first-order logic FO on every infinite class of ordered finite structures. Informally, the conjecture expresses an inherent limitation of first-order logic to capture polynomial-time computations on finite structures, no-matter how rich the combinatorial nature of the structures is. The question remains open, and it is known that any way of solving it will have important consequences in Complexity Theory. A refutation would imply that $\mathbf{P} \neq \mathbf{PSPACE}$ [12], and a proof would imply that $\mathbf{LINH} \neq \mathbf{E}$ [13]. Here, **LINH** is the linear-time hierarchy of Wrathall [35], and **E** is the usual complexity class that consists of all languages that are accepted by deterministic Turing machines in time $2^{O(n)}$.

There is a special case of the conjecture, singled out by Gurevich, Immerman, and Shelah [15], that is of particular interest. Namely, it is unknown whether LFP collapses to FO on the class of all finite structures of the form $(\{0,\dots,n-1\},\leq,\mathrm{BIT})$, where $\leq$ is the usual linear ordering, and BIT is the binary relation that consists of all pairs (p,q) of natural numbers such that the p-th bit in the binary expansion of q is one. As pointed out in [15], the collapse happens if and only if **DLOGTIME**-uniform $\mathbf{AC}^0$ = **P**-uniform $\mathbf{AC}^0$ (see Section 2 for definitions), or equivalently, if and only if $\mathbf{LINH} = \mathbf{E}$. Motivated by this interesting connection, Atserias and Kolaitis [3] investigated the difficulty of settling this special case of the Ordered Conjecture. Their approach is further motivated by the existence of a well-known isomorphism between $(\mathbb{N},\mathrm{BIT})$ and $(V_\omega,\in)$ (see [6]), where V_ω is the class of all hereditarily finite sets; that is, $V_\omega = \bigcup_{n\geq 0} V_n$, where $V_{n+1} = \mathcal{P}(V_n)$ and $V_0 = \emptyset$. The Ackermann bijection $e : \mathbb{N} \to V_\omega$ defined for every $n \in \mathbb{N}$ as

$$e(n) = \{e(m) : \text{the } m\text{-th bit of } n \text{ is one}\},$$

is the aforementioned isomorphism. Furthermore, by exploiting this mapping of BIT into $\in$, Dawar, Doets, Lindell and Weinstein [11] showed the somewhat surprising result that the standard linear order is first-order definable from the BIT predicate alone. Hence, the question translates into whether LFP collapses to FO on the class $\mathcal{BFR} = \{(\{e(0),\dots,e(n-1)\},\in) : n \geq 0\}$. In view of the Ackermann bijection, we identify the structures $(\{0,\dots,n-1\},\mathrm{BIT})$ and $(\{e(0),\dots,e(n-1)\},\in)$, here and in the future, and thus use the notation $\mathbf{BIT}_n = (\{0,\dots,n-1\},\in)$.

This *set-theoretic* framework led to the study of the fixed-points of the Δ_0 formulas of set theory, also called bounded formulas. These are the formulas all of whose quantifiers are of the form $(\exists x \in y)$ and $(\forall x \in y)$ (Sazonov has studied the fixed-points of bounded formulas in the context of definability on $(V_\omega,\in)$, rather than in the context of uniform definability on finite structures; see [30] for a survey). It was proved by Atserias and Kolaitis [3] that if the fixed-points of positive Δ_0 formulas $\mathrm{LFP}(\Delta_0)$ were first-order definable on $\mathcal{BFR}$, then

$\mathbf{P} \subseteq \mathbf{LINH}$ and so $\mathbf{P} \neq \mathbf{PSPACE}$. Thus, settling whether LFP(Δ_0) collapses is already a difficulty question. Nonetheless, the authors were able to show that the fixed-points of the so-called *restricted* Δ_0 formulas were indeed first-order definable, and so were the fixed-points of all unary and binary Δ_0 formulas. Finally back to complexity issues, they showed that the number of times that a positive Δ_0 formula has to be iterated until its fixed-point is reached in a finite structure, its closure function, is bounded by a polylogarithm of the cardinality of the structure on a small subclass of $\mathcal{BFR}$. As a consequence, these fixed-points are computable in **NC** on this class.

1.2 Main results

The isomorphism mapping BIT to the membership relation $\in$ constitutes a good source of inspiration to obtain results that explain the expressive power of first-order logic and fixed-point logic when strong built-in relations are available. The results in [11] and [3] are good examples. Moreover, the set-theoretic framework provides new concepts to consider, such as Δ_0 formulas, and new techniques to apply, such as absoluteness arguments. However, the complexity aspects of LFP(Δ_0) were not completely studied in [3], and we feel that the results of the present paper complete this study.

The first result of this paper is the extension of the last result in [3] to arbitrary classes of finite structures with built-in membership (BIT) relation. That is, we show that the closure functions of Δ_0 formulas are bounded by a polylogarithm of the cardinality of the universe of any arbitrary finite structure with built-in membership relation. Moreover, we observe that this implies that LFP(Δ_0) is computable in **DPOLYLOGTIME** (and not simply in **NC**) on any arbitrary class of finite structures with built-in membership relation. Then we focus back to the class $\mathcal{BFR}$. We observe that on this particular class, LFP(Δ_0) is even in (non-uniform) $\mathbf{AC}^0$ for some trivial reasons. The interesting question is then: Which uniform version of $\mathbf{AC}^0$ is captured by FO + LFP(Δ_0), the first-order closure of LFP(Δ_0)? Our second main result is the answer to this question: on $\mathcal{BFR}$, the logic FO + LFP(Δ_0) captures **DPOLYLOGTIME**-uniform $\mathbf{AC}^0$ which in turn, coincides with $\mathbf{LH}^{\mathbf{P}}$; the logarithmic-time hierarchy of Sipser with an oracle to **P**. As a corollary we obtain an exact characterization of the complexity-theoretic difficulties of showing FO + LFP(Δ_0) = FO on $\mathcal{BFR}$. We show that the collapse is equivalent to $\mathbf{P} \subseteq \mathbf{LINH}$. Note that **LINH** coincides with the rudimentary languages **RUD** [35] introduced by Smullyan [32].

We then consider the descriptive complexity of FO + LFP(Δ_0) on arbitrary classes of finite structures with built-in membership relation. Somewhat surprisingly, we are only able to provide an exact answer in the case that the underlying vocabulary of the class of structures is unary (on classes of words with built-in membership relation). In that case, FO + LFP(Δ_0) still captures **DPOLYLOGTIME**-uniform $\mathbf{AC}^0$. For higher arities, however, we are only able to compare the relative expressive power of FO + LFP(Δ_0) and FO with a complexity-theoretic question. We show that if $\mathbf{P} \subseteq \mathbf{RUD}_{n^{1/r}}$, then FO + LFP($\Delta_0$) collapses to FO on any arbitrary class of finite structures with built-in membership relation over a vocabulary of arity at most r. The class $\mathbf{RUD}_{n^{1/r}}$ was introduced by Jones [21] as a natural subclass of the rudimentary languages $\mathbf{RUD} = \mathbf{RUD}_n$. A result of Allender and Gore [1] implies that $\mathbf{RUD}_{n^\epsilon}$ coincides with $\mathrm{ATIME}(O(n^\epsilon), O(1))$ for every $\epsilon \in (0,1]$. Here, $\mathrm{ATIME}(t(n), a(n))$ is the class of languages accepted by alternating Turing machines in time $t(n)$ and $a(n)$ alternations. Moreover, as mentioned by Allender and Gore, $\mathbf{RUD}_{n^\epsilon}$ contains complete problems of each level of the polynomial-time hierarchy **PH** [33, 35].

It is interesting that **DPOLYLOGTIME**-uniform $\mathbf{AC}^0$ comes out of our results as a natural complexity class (we note that polylogtime uniformity has been considered at least once in the past by Allender and Gore [2], although in a completely different context). The reason amounts to a connection with the problem of the uniformity of Boolean circuits for integer division, an interesting issue that has received a good deal of attention [7, 28, 23, 19]. See the end of Section 5 for more details. Finally, it is obvious that our objects of study are intimately related to questions about the rudimentary languages, a well-studied topic [32, 8, 20, 35, 27, 1]. We point out that the rudimentary languages, and the techniques related to them, have been revisited very recently by Fortnow [14], and Lipton and Viglas [24], to obtain significant progress in some important open problems in Complexity Theory.

2 Preliminaries

Logic. Let $\sigma = \{R_1, \ldots, R_s\}$ be a finite relational vocabulary, and let $\mathbf{M} = (M, R_1^{\mathbf{M}}, \ldots, R_s^{\mathbf{M}})$ be a finite structure over σ. We will always identify the universe of $\mathbf{M}$, denoted M, with the initial segment of the natural numbers of cardinality $|M|$; thus, $M = \{0, \ldots, |M|-1\}$. Let $R = (R_1, R_2, \ldots)$ be a sequence of k-ary relations such that $R_n \subseteq \{0, \ldots, n-1\}^k$. Let C be a class of finite structures for $\sigma \cup \{R\}$, with $R \notin \sigma$. We say that C is a class of finite structures over σ with built-in R-relation if and only if, for every $\mathbf{M} \in C$, we have that $R^{\mathbf{M}} = R_{|M|}$. Notice that the built-in relation only depends on the cardinality of the structure.

Least fixed-point logic FO + LFP is the extension of first-order logic FO obtained by augmenting the syntax with a new formula $\mathrm{LFP}_{\overline{x},X}\,\varphi(x_1, \ldots, x_k, X)$, for every first-order formula φ positive in the k-ary relation variable X. The meaning of $\mathbf{M} \models (\mathrm{LFP}_{\overline{x},X}\varphi)[\overline{a}]$ is that $\overline{a} \in I_\varphi(\mathbf{M})$, where $I_\varphi(\mathbf{M})$ is the least fixed-point of the monotone operator defined by φ on $\mathbf{M}$. We let $I_\varphi^m(\mathbf{M})$ be the m-th stage, that is, $I_\varphi^m(\mathbf{M}) = \{\overline{a} \in M^k : \mathbf{M} \models \varphi[\overline{a}, \bigcup_{m'<m} I_\varphi^{m'}(\mathbf{M})]\}$. It is known that FO + LFP is closed under nested applications of the least fixed-point operator (see [17, 16]).

We let $\mathrm{LFP}(\Delta_0)$ be the class of formulas of the form $\mathrm{LFP}_{\overline{x},X}\,\varphi(x_1, \ldots, x_k, X)$, where φ is a Δ_0 formula positive in k-ary relation variable X. Observe that first-order parameters are not allowed, and neither is the nesting of fixed-point operators. We let $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ denote the closure of $\mathrm{LFP}(\Delta_0)$ under all first-order connectives and quantification.

Complexity. For every natural number n, we let $\log n$ denote the length of the shortest binary representation of n. If we wish to use the true base-two logarithm, we use the notation $\log_2(n)$; thus, $\log n = \lfloor \log_2(n) \rfloor + 1$. We identify natural numbers with their shortest binary representation. However, for every $m \in \{0, \ldots, n-1\}$, we let $\mathrm{b}_n(m)$ denote the unique binary representation of length $\log(n-1)$ (padded with leading zeros if necessary).

Our model of computation is the oracle alternating multitape Turing machine with random access to the input. This model, originally defined by Ruzzo [29] and used by Barrington, Immerman and Straubing [5], Buss [9], and Sipser [31] among others, is a modification of the model of Chandra, Kozen and Stockmeyer [10] to allow sublinear time-bounds. These machines are equipped with an address tape on which to write a number in binary. When the machine enters a distinguished state with a number p written on its address tape, the head of the input tape jumps, in one step, to the p-th leftmost cell of the tape. Strictly speaking, the definition of Ruzzo [29] is slightly different from ours, but standard simulation arguments show that both models have the same computing power with only a constant factor loss in time or number of alternations (see [5] and [9] for example). In the case of deterministic machines, our model is slightly more robust, but this will not affect the generality of the results.

Finite structures are encoded as words over the alphabet $\{0, 1, \#\}$ according to the following convention. For every relation symbol $R_i \in \sigma$ of arity r, we let $\chi(R_i^{\mathbf{M}})$ be the characteristic sequence of $R_i^{\mathbf{M}}$. That is, $\chi(R_i^{\mathbf{M}}) = a_0 a_1 \ldots a_{n^r-1}$, where $a_m \in \{0, 1\}$, and $a_m = 1$ if and only if $(m_{r-1}, \ldots, m_0) \in R_i^{\mathbf{M}}$ where $(m_{r-1}, \ldots, m_0)$ is the n-ary representation of m. Then, the encoding of $\mathbf{M}$ is just

$$\langle \mathbf{M} \rangle = 1^n \# \chi(R_1^{\mathbf{M}}) \# \ldots \# \chi(R_s^{\mathbf{M}}).$$

We extend the encoding to include individuals as follows. For every $a_1, \ldots, a_k \in M$, let $\langle \mathbf{M}, a_1, \ldots, a_k \rangle = \langle \mathbf{M} \rangle \# \mathrm{b}_n(a_1) \# \ldots \# \mathrm{b}_n(a_k)$. Let C be a class of finite structures, and let Q be a k-ary query on C. We say that Q is computable in a complexity class $\mathcal{C}$ on C if there exists a language $L \in \mathcal{C}$ such that for every $\mathbf{M} \in C$ and $a_1, \ldots, a_k \in M$, we have that $(a_1, \ldots, a_k) \in Q(\mathbf{M})$ if and only if $\langle \mathbf{M}, a_1, \ldots, a_k \rangle \in L$. We say that a k-ary built-in relation $R = (R_1, R_2, \ldots)$ is computable in a complexity class $\mathcal{C}$ if there exists a language $L \in \mathcal{C}$ such that for every n and $a_1, \ldots, a_k \in \{0, \ldots, n-1\}$, we have that $(a_1, \ldots, a_k) \in R_n$ if and only if $1^n \# \mathrm{b}_n(a_1) \# \ldots \# \mathrm{b}_n(a_k) \in L$. When considering Boolean circuits, we are forced to restrict ourselves to the binary alphabet $\{0, 1\}$. We fix then an homomorphism $h : \{0, 1, \#\}^* \to \{0, 1\}^*$ in a standard way: put $h(0) = 00$, $h(1) = 11$ and $h(\#) = 01$ (see Section 4 for more details).

3 General facts about bounded formulas

Recall that the transitive closure of a set a, denoted by $\mathrm{TC}(a)$, is defined inductively as follows: $\mathrm{TC}(a) = \bigcup \{\mathrm{TC}(b) : b \in a\}$. The reflexive transitive closure of a, denoted by $\mathrm{RTC}(a)$, is $\{a\} \cup \mathrm{TC}(a)$. Our first Lemma says that the satisfiability of a Δ_0 formula only depends on the reflexive transitive closure of its arguments. Given a first-order formula $\varphi(x_1, \ldots, x_n)$ with free variables among $x_1, \ldots, x_n$, we let $F(\varphi)$ be the set of *indices* of the free variables of φ.

Lemma 1. *Let σ be a relational vocabulary, let $\mathbf{M}$ be a structure for $\sigma \cup \{\in\}$ with built-in membership relation, and let $\varphi(x_1, \ldots, x_s, X)$ be a Δ_0 formula over $\sigma \cup \{\in, X\}$, where X is a k-ary relation variable. For every $A \subseteq M^k$, and every tuple $\overline{a} = (a_1, \ldots, a_s) \in M^s$, we have that $\mathbf{M} \models \varphi[\overline{a}, A]$ if and only if $\mathbf{M} \models \varphi[\overline{a}, A \cap (\bigcup\{\mathrm{RTC}(a_i) : i \in F(\varphi)\})^k]$.*

Proof: We proceed by induction on the construction of φ. The base cases are trivial, and so is the case in which φ is of the form $\neg\psi$. Suppose that φ is of the form $\psi_1 \wedge \psi_2$. Let $B = \bigcup\{\mathrm{RTC}(a_i) : i \in F(\varphi)\}$ and $B_j = \bigcup\{\mathrm{RTC}(a_i) : i \in F(\psi_j)\}$ for $j = 1, 2$. Then, $\mathbf{M} \models \varphi[\overline{a}, A \cap B^k]$ if and only if $\mathbf{M} \models \psi_j[\overline{a}, A \cap B^k]$ for $j = 1, 2$. By induction hypothesis, this is equivalent to $\mathbf{M} \models \psi_j[\overline{a}, A \cap B^k \cap B_j^k]$ for $j = 1, 2$. Since $F(\psi_j) \subseteq F(\varphi)$, this is equivalent to $\mathbf{M} \models \psi_j[\overline{a}, A \cap B_j^k]$ for $j = 1, 2$. By induction hypothesis again, this is equivalent to $\mathbf{M} \models \psi_j[\overline{a}, A]$ for $j = 1, 2$, and therefore to $\mathbf{M} \models \varphi[\overline{a}, A]$. Suppose next that φ is of the form $(\exists x_i \in x_j)\psi$. Let $B = \bigcup\{\mathrm{RTC}(a_i) : i \in F(\varphi)\}$. In this case, $\mathbf{M} \models \varphi[\overline{a}, A \cap B^k]$ if and only if there is some $a \in M$ such that $a \in a_j$ and $\mathbf{M} \models \psi[\overline{b}, A \cap B^k]$, where $\overline{b} = (a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_s)$. Let $B(\overline{b}) = \bigcup\{\mathrm{RTC}(b_l) : l \in F(\psi)\}$. Therefore, by induction hypothesis, $\mathbf{M} \models \varphi[\overline{a}, A \cap B^k]$ if and only if there is some $a \in M$ such that $a \in a_j$ and $\mathbf{M} \models \psi[\overline{b}, A \cap B^k \cap B(\overline{b})^k]$. Since for every $a \in a_j$ we have that $\mathrm{RTC}(a) \subseteq \mathrm{RTC}(a_j)$, it is the case that $B(\overline{b}) \subseteq B$. Consequently, $\mathbf{M} \models \varphi[\overline{a}, A \cap B^k]$ if and only if there is some $a \in M$ such that $a \in a_j$ and $\mathbf{M} \models \psi[\overline{b}, A \cap B(\overline{b})^k]$, and by induction hypothesis again, $\mathbf{M} \models \psi[\overline{b}, A]$, as required. □

For every first-order formula $\varphi(x_1, \ldots, x_k, X)$ positive in the k-ary relation symbol X, we let $\mathrm{cl}_\varphi(\mathbf{M})$ denote the closure ordinal of φ in $\mathbf{M}$; that is, $\mathrm{cl}_\varphi(\mathbf{M})$ is the minimum ordinal α such that $I_\varphi^\alpha(\mathbf{M}) = \bigcup_{\alpha' < \alpha} \mathbf{I}_\varphi^{\alpha'}(\mathbf{M})$ [26]. Since the reflexive transitive closure of a finite set is relatively small, Lemma 1 allows us to put polylogarithmic bounds on the closure functions of Δ_0 formulas. This result extends Theorem 4 in [3] to the case of $\mathcal{BFR}$, and in fact, to arbitrary classes of finite structures with built-in membership relation.

Theorem 1. *Let σ be a relational vocabulary, and let $\varphi(x_1, \ldots, x_k, X)$ be a Δ_0 formula over $\sigma \cup \{\in, X\}$ that is positive in the k-ary relation variable X. Then,*

$$\mathrm{cl}_\varphi(\mathbf{M}) \leq (\log(|M| - 1) + k)^k$$

for every finite structure $\mathbf{M}$ over $\sigma \cup \{\in\}$ with built-in membership relation.

Proof: Put $t = (\log(|M|-1)+k)^k$, and assume for contradiction that $\mathrm{cl}_\varphi(\mathbf{M}) > t$. Let $\overline{a}_0 = (a_{0,1}, \ldots, a_{0,k}) \in I_\varphi(\mathbf{M})$ be such that $|\overline{a}_0| > t$, where $|\overline{a}|$ denotes the minimal m such that $\overline{a} \in I_\varphi^m(\mathbf{M})$ if $\overline{a} \in I_\varphi(\mathbf{M})$, and ∞ if $\overline{a} \notin I_\varphi(\mathbf{M})$. In the following, let I^m be an abbreviation for $I_\varphi^m(\mathbf{M})$. We build a sequence $\overline{a}_0, \overline{a}_1, \ldots, \overline{a}_t$ such that $|\overline{a}_i| = |\overline{a}_0| - i$, and $\overline{a}_i \in S^k$ for every $i = 0, \ldots, t$, where $S = \{0, \ldots, \log(|M|-1) - 1\} \cup \{a_{0,1}, \ldots, a_{0,k}\}$. This will prove the theorem since the cardinality of S^k is at most t.

For every $\overline{a} = (a_1, \ldots, a_k) \in M^k$, let $S(\overline{a})$ denote the set $\{0, \ldots, \log(|M| - 1)\} \cup \{a_1, \ldots, a_k\}$. Observe that $\bigcup\{\mathrm{RTC}(a_i) : i \in F(\varphi)\} \subseteq S(\overline{a})$ since every element in $\mathrm{TC}(a_i)$ is a *bit* position of an element in $\{0, \ldots, |M| - 1\}$. Assuming $\overline{a}_i = (a_{i,1}, \ldots, a_{i,k})$ is already defined, we define $\overline{a}_{i+1} = (a_{i+1,1}, \ldots, a_{i+1,k})$. Let $m = |\overline{a}_i|$. Then, $\mathbf{M} \models \varphi[\overline{a}_i, I^{m-1}]$. Lemma 1 and monotonicity imply that $\mathbf{M} \models \varphi[\overline{a}_i, I^{m-1} \cap S(\overline{a}_i)^k]$. Observe that since $\overline{a}_i \in S(\overline{a}_0)^k$ by assumption, we have that $S(\overline{a}_i) \subseteq S(\overline{a}_0)$. Now let us consider two cases: (i) $I^{m-1} \cap S(\overline{a}_0)^k \subseteq I^{m-2}$, or (ii) $I^{m-1} \cap S(\overline{a}_0)^k \not\subseteq I^{m-2}$. In case (i) we have that $\mathbf{M} \models \varphi[\overline{a}_i, I^{m-2}]$ by monotonicity. Hence, $\overline{a}_i \in I^{m-1}$ which contradicts the minimality of $m = |\overline{a}_i|$. In case (ii), there must exist some $\overline{a}_{i+1} \in I^{m-1} \cap S(\overline{a}_0)^k$ that does not belong to I^{m-2}. Observe that $|\overline{a}_{i+1}| = |\overline{a}_i| - 1$, and $\overline{a}_{i+1} \in S(\overline{a}_0)^k$ as required. This completes the proof of the theorem. □

The polylogarithmic bounds on the closure functions, together with a result of Immerman [18], imply that every query that is definable as the fixed-point of a Δ_0 formula is computable in **NC**, the parallel complexity class. However, we can keep the machine sequential as noted in the following

Lemma 2. *Let σ be a relational vocabulary, let C be a class of finite structures over σ with built-in membership relation, and let $\varphi(x_1, \ldots, x_k, X)$ be a Δ_0 formula that is positive in the k-ary relation variable X. Then, the query on C defined by the formula* $(\mathrm{LFP}_{\overline{x},X}\,\varphi)$ *is computable in* **DPOLYLOGTIME** *on C.*

Proof: The idea is that the standard fixed-point computation will only take a polylogarithmic number of iterations by Theorem 1, and each iteration is computable in polylogarithmic-time because φ is a Δ_0 formula. More precisely, on input $\langle \mathbf{M}, a_1, \ldots, a_k \rangle$, the polylogarithmic-time Turing machine will proceed as follows. The machine first determines the cardinality of M, say n. To this end, it determines the length m of the input in $O(\log m)$ steps using its random access to the input (see [5] for this trick), and then it executes a straightforward computation to extract n from m (here we use the fact that our encodings are carefully chosen so that their length is determined by the cardinality of $\mathbf{M}$, the signature of σ, and k). Let $B = \{0, \ldots, \log(n-1) - 1\} \cup \{a_1, \ldots, a_k\}$. The machine will keep, in a separate tape, an encoding of a k-ary relation on B; this will require $O((\log n)^k)$ bits of information. Then, it starts a loop that is to be repeated $(\log(n-1) + k)^k$ times. In each iteration, the machine cycles through all k-tuples $(b_1, \ldots, b_k)$ in B^k, and evaluates $\mathbf{M} \models \varphi[b_1, \ldots, b_k, R]$ where R is the k-ary relation encoded in the separate tape. Atomic formulas from σ are resolved by random access to the input, and atomic formulas of the form $X(u_1, \ldots, u_k)$ are resolved by accessing the position of tuple $(u_1, \ldots, u_k)$ in the encoding of R. Observe that each relevant tuple $(u_1, \ldots, u_k)$ will be available since $\mathbf{M} \models \varphi[b_1, \ldots, b_k, R]$ if and only if

$$\mathbf{M} \models \varphi[b_1, \ldots, b_k, R \cap (\bigcup\{\mathrm{RTC}(b_i) : i \in F(\varphi)\})^k],$$

and $\bigcup\{\mathrm{RTC}(b_i) : i \in F(\varphi)\} \subseteq B$ for every $(b_1, \ldots, b_k) \in B^k$, since every element of $\mathrm{TC}(a_i)$ is a *bit* position of an element in $\{0, \ldots, n-1\}$. For the same reason, each quantifier is bounded by some b_i, and therefore, the variable it bounds ranges over at most $\log(n-1)$ elements of the universe. Hence, the computation can be done in time $O((\log n)^r)$ where r depends on the number of quantifiers of φ. When the evaluation of $\mathbf{M} \models \varphi[b_1, \ldots, b_k, R]$ is complete, the machine updates accordingly the position corresponding to tuple $(b_1, \ldots, b_k)$ in the encoding of R. Finally, the machine will only have to check whether the tuple $(a_1, \ldots, a_k)$ belongs to R at the end of the loop. □

4 Fixed-points of bounded formulas on $\mathcal{BFR}$

A closer examination of Lemma 2 in the case of $\mathcal{BFR}$ reveals that LFP(Δ_0)-definable queries are also computable in (non-uniform) $\mathbf{AC}^0$; the reason is that they only depend on $O(\log n)$ bits of the input (in fact, the relevant part of the input is that short already). The interesting question at this point is the following: which uniform version of $\mathbf{AC}^0$ is captured by FO+LFP(Δ_0) on $\mathcal{BFR}$? Our next theorem is the answer to this question. Before stating the result, we need some definitions.

Let $C = (C_1, C_2, \ldots)$ be a sequence of boolean circuits, and let s_n be a bound on the size of C_n. Thus, gates in C_n may be numbered in $\{0, \ldots, s_n - 1\}$. The direct connection language of C (see [5]) is the set of words of the form $1^n\#\mathrm{b}_{s_n}(a)\#\mathrm{b}_{s_n}(b)\#t$, where gate b is an input to gate a, and the type of gate b is $t \in \{0, 1, 2, 3\}$. Here, $t = 0$ means that b is an AND gate, $t = 1$ means that b is an OR gate, $t = 2$ means that b is an positive input, and $t = 3$ means that b is a negated input. If $\mathcal{C}$ is a complexity class, we say that C is $\mathcal{C}$-uniform if there exists a language in $L \in \mathcal{C}$ such that for every word w of the form $1^n\#a\#b\#t$ we have that $w \in L$ if and only if $w \in \mathrm{DCL}(C)$. The class $\mathcal{C}$-uniform $\mathbf{AC}^0$ is the class of all languages that are accepted by a $\mathcal{C}$-uniform, polynomial-size, bounded-depth, family of circuits (for languages $L \subseteq \Sigma^*$ with $\Sigma \neq \{0, 1\}$, we say that L is accepted by a family of circuits C if $h(L)$ is accepted by C for some fixed homomorphism $h : \Sigma^* \to \{0, 1\}^*$ [5]).

Theorem 2. *Let Q be a query on $\mathcal{BFR}$. The following are equivalent:*

1. *Q is computable in* $\mathbf{LH}^{\mathbf{P}}$ *on $\mathcal{BFR}$,*
2. *Q is computable in* **DPOLYLOGTIME**-*uniform* $\mathbf{AC}^0$ *on $\mathcal{BFR}$,*
3. *Q is definable in* FO + LFP(Δ_0) *on $\mathcal{BFR}$.*

Proof: We close a cycle of implications. We first show that (i) implies (ii). Assume that Q is computable in $\mathbf{LH}^A$ for some $A \in \mathbf{P}$. We may assume $A \subseteq \{0, 1\}^*$. For every n, let $F_n(x_1, \ldots, x_n)$ be the following DNF-formula

$$\bigvee_{\overline{a} \in A \cap \{0,1\}^n} \left(\bigwedge_{a_i = 1} x_i \wedge \bigwedge_{a_i = 0} \neg x_i \right).$$

Observe that $F_n(a_1, \ldots, a_n)$ is true if and only if the word $a_1 \ldots a_n$ belongs to A. The sequence $(F_1, F_2, \ldots)$, interpreted as a sequence of depth-two circuits, is exponential-size in n, but $\mathbf{P}$-uniform (the words of its direct connection language are of the form $1^n\#a\#b\#t$ with $a, b \in \{0,1\}^{O(n)}$, and deciding membership can be done in polynomial-time since $A \in \mathbf{P}$). We now build a **DPOLYLOGTIME**-uniform family of $\mathbf{AC}^0$ circuits to compute Q. Let M be an oracle alternating Turing machine witnessing that Q is computable in $\mathbf{LH}^A$, and assume that M queries its oracle at most once in each computation path (this is a standard trick in alternating machines; it consists of existentially guessing the answers, write them down on a separate tape together with the nondeterministic branch taken at each step, and at the end of the computation, universally branch to check the correctness of every guess by deterministically resimulating the computation path until the challenged query is asked). Let $c \log n$ be a bound on the running-time of M on inputs of length n. Observe that the length of each oracle query is bounded by $c \log n$ too. As in [5], we may see the computation trees of M as a **DLOGTIME**-uniform family of $\mathbf{AC}^0$ circuits, except for the oracle queries, which may be resolved by **DPOLYLOGTIME**-uniform $\mathbf{AC}^0$ circuits; namely, we let queries of length $m \leq c \log n$ be resolved by the circuit F_m above (exponential-size in $m \leq c \log n$ is polynomial-size in n, and polynomial-time uniformity for length $m \leq c \log n$ is polylogarithmic-time uniformity for length n). It follows that Q is computable in **DPOLYLOGTIME**-uniform $\mathbf{AC}^0$.

We see that (ii) implies (iii). Let Q be computable in **DPOLYLOGTIME**-uniform $\mathbf{AC}^0$ (recall the convention established just before the statement of the theorem). It is well-known [4, 25] that Q is then first-order definable with an additional built-in relation $R = (R_1, R_2, \ldots)$ that is computable in polylogarithmic-time. We show how to replace every occurrence of this built-in relation by a formula of FO + LFP(Δ_0). For every n and $\overline{a} = (a_1, \ldots, a_k) \in \{0, \ldots, n-1\}^k$, let $\mathbf{M}_{\overline{a}} = (\{0, \ldots, \log(n-1) - 1\}, \in, P_1^n, \ldots, P_k^n)$, where $P_i^n = \{m : m \in a_i\}$. Since R is a built-in relation computable in polylogarithmic-time, the language

$$\{1^n\#\mathrm{b}_n(a_1)\#\ldots\#\mathrm{b}_n(a_k) : (a_1, \ldots, a_k) \in R_n,\ n \geq 1\}$$

is decidable in polylogarithmic-time on inputs of the appropriate form. A simple unpadding argument shows then that the language $\{\langle\mathbf{M}_{\overline{a}}\rangle : \overline{a} \in \bigcup_{n\geq 1} R_n\}$ is in $\mathbf{P}$ (the length of $\langle\mathbf{M}_{\overline{a}}\rangle$ is logarithmic in the length of $\langle\{0, \ldots, n-1\}, a_1, \ldots, a_k\rangle$). Hence, by the Immerman-Vardi Theorem, the boolean query $Q = \{\mathbf{M}_{\overline{a}} : \overline{a} \in \bigcup_{n\geq 1} R_n\}$ is definable in least fixed-point logic on the class of all structures of the form $\mathbf{M}_{\overline{a}}$. We may even assume that Q is definable by a sentence of the form $(\mathrm{LFP}_{\overline{x},X}\varphi)(\overline{0})$ in which φ is a first-order formula, and 0 is a constant for zero. Let $\varphi'(y, z, p_1, \ldots, p_k, \overline{x}, X)$ be the first-order formula over the vocabulary $\{\in\}$ that results from the following substitution in φ: replace each occurrence of an atomic formula of the form $P_i(u)$ by $u \in p_i$; replace each atomic formula of the form $X(\overline{u})$ by $X(y, z, p_1, \ldots, p_k, \overline{u})$; and replace each subformula of the form $(\exists u)(\psi)$ by $(\exists u \in y)(\psi') \vee (\exists u \in z)(\psi')$, where ψ' is the result of applying recursively the substitutions. Clearly, φ' is a Δ_0 formula. Moreover, it is not hard to see that for every $a_1, \ldots, a_k \in \{0, \ldots, n-1\}$, we have that $\mathbf{M}_{\overline{a}} \models (\mathrm{LFP}_{\overline{x},X}\varphi)(\overline{0})$ if and only if

$$(\{0, \ldots, \log(n-1) - 1\}, \in) \models (\mathrm{LFP}_{y,z,\overline{p},\overline{x},X}\varphi')(r, s, a_1, \ldots, a_k, \overline{0}),$$

where s is the largest power of two in the universe, and $r = s - 1$ (observe that the binary representations of s and r are dual words; that is, $j \in s$ if and only if $j \notin r$ for every $j \leq \log(\log(n-1)-1)$). Since r and s are first-order definable with $\in$, we have shown that R is uniformly definable on $\mathcal{BFR}$ by a sentence of $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$.

It remains to see that (iii) implies (i). Let $\varphi(x_1, \ldots, x_s)$ be a formula witnessing that Q is definable in $\mathrm{FO}+\mathrm{LFP}(\Delta_0)$. Without loss of generality, we may assume the following normal form for φ:

$$(Q_1 y_1) \cdots (Q_r y_r)(\bigwedge_{i=1}^{s} (\psi_{i,1} \vee \ldots \vee \psi_{i,t} \vee \neg\psi_{i,t+1} \vee \neg\psi_{i,u})),$$

where each Q_i is $\exists$ or $\forall$, and each $\psi_{i,j}$ is either an atomic formula, or a formula of the form $(\mathrm{LFP}_{\overline{z},Z}\theta)(z_1, \ldots, z_s)$, with θ a Δ_0 formula. For every i, j, let $Q_{i,j}$ be the query on $\mathcal{BFR}$ defined by $\psi_{i,j}$, and let A be the following language over the alphabet $\{0, 1, \#\}$:

$$\{n\#\mathrm{b}_n(b_1)\#\ldots\#\mathrm{b}_n(b_s)\#i\#j : (b_1, \ldots, b_s) \in Q_{i,j}(\mathbf{BIT}_n)\}.$$

This language will be our oracle set (that it belongs to $\mathbf{P}$ will be shown later). An alternating Turing machine with oracle A may simulate φ as indicated next. On input $\langle \mathbf{BIT}_n, a_1, \ldots, a_k \rangle$ where $a_1, \ldots, a_k \in \{0, \ldots, n-1\}$, the machine behaves as follows. First, it computes n. To this end, it existentially guesses the position of the leftmost # in the input, and universally branches to check that every smaller position contains a symbol other than #. Then, following the alternation pattern of the quantifier prefix of φ, the machine existentially or universally guesses r words $w_1, \ldots, w_r$ of length $\log(n-1)$ each. The i-th word w_i is meant to be the binary representation of an element $b_i \in \{0, \ldots, n-1\}$ that is to interpret the first-order variable y_i. The machine proceeds then to evaluate each atomic formula $\psi_{i,j}$ as follows. Assume $\psi_{i,j} = \psi_{i,j}(z_1, \ldots, z_s)$, where each variable z_k is either an x_l or a y_l. The machine will write an oracle query of the form $n\#d_1\#\ldots\#d_s\#i\#j$, where $d_k = \mathrm{b}_n(b_l)$ if $z_k = y_l$, and $d_k = \mathrm{b}_n(a_l)$ if $z_k = x_l$. Observe that the length of this query is $O(\log n)$, and is easy to recover from the input (existentially guess each $\mathrm{b}_n(a_l)$ and universally branch to check that all guesses match the input). Clearly, the answer to this query is yes if and only if $\mathbf{BIT}_n \models \psi_{i,j}[d_1, \ldots, d_s]$ by the definition of the oracle set A.

All it remains to show is that the language A belongs to $\mathbf{P}$. This is fairly easy. If $\psi_{i,j}$ is an atomic formula, there is almost nothing to see: equalities are checked at once, and atomic formulas of the form $z_i \in z_j$ are also straightforward to check. If $\psi_{i,j}$ is a formula of the form $(\mathrm{LFP}_{\overline{z},Z}\theta)(z_1, \ldots, z_s)$ with θ being a Δ_0 formula, then the query it defines is computable in **DPOLYLOGTIME** on $\mathcal{BFR}$ by Lemma 2. Therefore, since the length of $n\#d_1\#\ldots\#d_s\#i\#j$ is logarithmic in the length of $\langle \mathbf{BIT}_n, d_1, \ldots, d_s \rangle$, a simple unpadding argument puts A in $\mathbf{P}$. $\Box$

As a corollary, we obtain a characterization of the question on whether all polynomial-time decidable languages are rudimentary. The relationship between

P and **RUD** remains unknown. It is known however that $\mathbf{NL} \subseteq \mathbf{RUD}$ [27], where **NL** is the class of languages accepted in nondeterministic logarithmic-space.

Corollary 1. *The following are equivalent:*

1. $\mathrm{FO} + \mathrm{LFP}(\Delta_0) \subseteq \mathrm{FO}$ *on* $\mathcal{BFR}$,
2. $\mathbf{P} \subseteq \mathbf{RUD}$,
3. $\mathbf{P} \subseteq \mathbf{LINH}$.

Proof: Since $\mathbf{RUD} = \mathbf{LINH} = \mathrm{ATIME}(O(n), O(1))$, it is enough to show that (i) and (iii) are equivalent. The implication from (i) to (iii) follows from Theorem 1 in [3]. For the other implication, assume that $\mathbf{P} \subseteq \mathbf{LINH}$, and let Q be a query on $\mathcal{BFR}$ that is definable by a $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ formula. By Theorem 2 we have Q is computable in $\mathbf{LH}^{\mathbf{P}}$, and so in $\mathbf{LH}^{\mathbf{LINH}}$ by hypothesis. Let M be an oracle alternating Turing machine witnessing that Q is computable in $\mathbf{LH}^{A}$ for some $A \in \mathbf{LINH}$, and let N be an alternating Turing machine witnessing that $A \in \mathbf{LINH}$. Since an oracle Turing machine running in logarithmic-time can only ask logarithmically long queries, oracle queries of M may be answered by N in logarithmic-time with respect to the input to M. The number of alternations being constant, it follows that Q is computable in **LH**. Hence, Q is first-order definable on $\mathcal{BFR}$. □

5 The presence of input predicates

The natural question at this point is what happens when input predicates, in addition to the membership (BIT) relation, are available. That is, we fix a relational vocabulary σ, and we wonder what is captured by $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ on classes of finite structures over σ with built-in membership relation. Somewhat surprisingly, we are only able to provide an exact answer in the case that σ is a unary vocabulary. In that case, the $\mathrm{LFP}(\Delta_0)$-definable queries still only depend on $O(\log n)$ bits of the input, and a similar argument as before goes through.

Theorem 3. *Let σ be a unary vocabulary, let C be a class of finite structures over σ with built-in membership relation, and let Q be a query on C. Then, the following are equivalent:*

1. *Q is computable in* $\mathbf{LH}^{\mathbf{P}}$ *on C,*
2. *Q is computable in* **DPOLYLOGTIME**-*uniform* $\mathbf{AC}^0$ *on C,*
3. *Q is definable in* $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ *on C.*

Proof: The proofs that (i) implies (ii), and that (ii) implies (iii), go through as in Theorem 2 essentially without change. The proof that (iii) implies (i) uses an argument similar to the one in the proof of Lemma 2. Recall from Lemma 1 that if φ is a Δ_0 formula, then $\mathbf{M} \models \varphi[a_1, \ldots, a_s, A]$ if and only if $\mathbf{M} \models \varphi[a_1, \ldots, a_s, A \cap (\bigcup\{\mathrm{RTC}(a_i) : i \in F(\varphi)\})^k]$. Iterated application of this lemma with each of the relation symbols of σ shows then that $\mathbf{M} \models \varphi[a_1, \ldots, a_s, A]$ if and only if

$$\mathbf{M} \cap B \models \varphi[a_1, \ldots, a_s, A \cap B^k],$$

where $B = \bigcup\{\mathrm{RTC}(a_i) : i \in F(\varphi)\}$, and $\mathbf{M} \cap B$ is the substructure of $\mathbf{M}$ generated by B. In turn, we remark that $B \subseteq \{0, \ldots, \log(|M| - 1) - 1\} \cup \{a_1, \ldots, a_s\}$ since each element of $\mathrm{TC}(a_i)$ is a *bit* position of an element in $\{0, \ldots, |M| - 1\}$. Moreover, a straightforward argument reveals that $\mathbf{M} \cap B'$ is an end-extension of $\mathbf{M} \cap B$, where $B' = \{0, \ldots, \log(|M| - 1) - 1\} \cup \{a_1, \ldots, a_s\}$. Hence, $\mathbf{M} \models \varphi[a_1, \ldots, a_s, A]$ if and only if $\mathbf{M} \cap B \models \varphi[a_1, \ldots, a_s, A \cap B^k]$, and by absoluteness, if and only if $\mathbf{M} \cap B' \models \varphi[a_1, \ldots, a_s, A \cap {B'}^k]$. With these observations in hand, we claim that:

Claim. If Q is definable in $\mathrm{FO}+\mathrm{LFP}(\Delta_0)$ on C, then Q is definable by a formula of $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ in which no relation symbol from σ appears within the scope of a fixed-point operator.

Proof: The main idea is that since every $\mathrm{LFP}(\Delta_0)$ formula will only depend on $O(\log n)$ bits of the input predicates by the remarks above (here is the crucial point where we use the fact that the vocabulary is unary), we can existentially quantify these bits outside the $\mathrm{LFP}(\Delta_0)$-formula, and pass them to it as input variables. Formally, the argument is as follows. Assume for simplicity that σ consists of a unique relation symbol R; the general case is as easy. Let φ be a formula defining Q on C. Replace each occurrence in φ of a subformula of the form $(\mathrm{LFP}_{\overline{x},X}\theta)(x_1, \ldots, x_k)$ with θ a Δ_0 formula, by the formula

$$\begin{aligned}&(\exists v)((r \in v \leftrightarrow R(r)) \wedge (\forall z \in s)(z \in v \leftrightarrow R(z)) \wedge \\ &\bigvee_{w \in \{0,1\}^k} (\bigwedge_{w_i = 1} R(x_i) \wedge \bigwedge_{w_i=0} \neg R(x_i) \wedge (\mathrm{LFP}_{v,\overline{x},X'}\theta^w)(v, \overline{x}))),\end{aligned}$$

where θ^w is the result of replacing each atomic formula of the form $R(u)$, with u a bound variable, by $u \in v$, each atomic formula of the form $R(x_i)$ by $x_i = x_i$, if $w_i = 1$, each atomic formula of the form $R(x_j)$ by $x_j \neq x_j$, if $w_j = 0$, and each atomic formula of the form $X(\overline{u})$ by $X'(v, \overline{u})$. Here, r and s are existentially quantified variables set to the largest power of two of the universe, and $r - 1$ respectively (observe that the binary representations of r and s are dual words). Observe that if v is a witness for the first-order variable of this formula, then its binary representation is encoding the first $\log(n - 1)$ bits of R. By the remarks preceding the claim, it is straightforward to check using standard absoluteness arguments that the modified formula is defining Q on C, as required.

The rest of the proof that (iii) implies (i) is now almost identical to the proof of Theorem 2. Namely, access to the input predicates is only required when simulating the first-order part of the formula, and the simulation of the $\mathrm{LFP}(\Delta_0)$-parts of the formula may be asked to an oracle set in $\mathbf{P}$. $\square$

Observe that the argument of Theorem 3 does not go through for vocabularies of higher arities. In the case of digraphs, for example, the reason is that there are $O((\log|M|)^2)$ significant bits (instead of $O(\log|M|)$) in the substructure $\mathbf{M}\cap\{0,\ldots,\log(|M|-1)\}$ of any digraph $\mathbf{M}$. Although we do not provide with an exact characterization of FO+LFP(Δ_0) for vocabularies of higher arities, we are able to compare the expressive power of FO+LFP(Δ_0) with a familiar complexity class. Recall from the introduction that $\mathbf{RUD}_{n^{1/r}} = \mathrm{ATIME}(O(n^{1/r}), O(1))$ (see Corollary 5 in [1]).

Theorem 4. *Let σ be a relational vocabulary of maximum arity r, and let C be the class of all finite structures over σ with built-in membership relation. If $\mathbf{P} \subseteq \mathbf{RUD}_{n^{1/r}}$, then $\mathrm{FO} + \mathrm{LFP}(\Delta_0) \subseteq \mathrm{FO}$ on C.*

Proof sketch: Assume $\mathbf{P} \subseteq \mathbf{RUD}_{n^{1/r}}$, and let Q be a query on C definable in FO+LFP(Δ_0). It is enough to show that Q is computable in $\mathbf{LH}$ on C. Even easier, it is enough to show that each FO+LFP(Δ_0)-formula can be evaluated in $\mathbf{LH}$ on the appropriate inputs. Let $\varphi(x_1,\ldots,x_k)$ be such a formula. Lemma 2 says that deciding whether $\mathbf{M} \models \varphi[a_1,\ldots,a_k]$ can be done in polylogarithmic-time in $|M|$. Moreover, the same absoluteness argument as in the proof of Theorem 3 reveals that $\mathbf{M} \models \varphi[a_1,\ldots,a_k]$ if and only if $\mathbf{M}\cap B' \models \varphi[a_1,\ldots,a_k]$, where $B' = \{0,\ldots,\log(|M|-1)-1\} \cup \{a_1,\ldots,a_k\}$. Since only $O((\log|M|)^r)$ bits are relevant in $\mathbf{M}\cap B'$, the same computation can can be carried over an unpadded input that only contains these bits. The computation time is now polynomial in the length of the (unpadded) input, and therefore, by hypothesis, the same language is decidable in $\mathrm{ATIME}(O(n^{1/r}), O(1)) = \mathbf{RUD}_{n^{1/r}}$ on the appropriate inputs. Since the length of these inputs is $O((\log|M|)^r)$, the alternating computation can be carried over the original inputs in time

$$O(((\log|M|)^r)^{1/r}) = O(\log|M|),$$

and still a constant number of alternations. That is, on the original inputs, the evaluation of φ can be done in $\mathbf{LH}$ as required. □

As mentioned in the introduction, Theorem 3 sets the link to an important problem related to the uniformity of circuits for integer division. Beame, Cook, and Hoover [7] showed that the problem of dividing two numbers can be computed by $\mathbf{P}$-uniform bounded fan-in, logarithmic-depth circuits ($\mathbf{NC}^1$). The result was improved by Reif [28] (see also [19]) who showed that the problem could be computed by $\mathbf{P}$-uniform unbounded fan-in, bounded-depth circuits with majority gates ($\mathbf{TC}^0$). However, it is not known whether the uniformity condition of their algorithm can be relaxed to $\mathbf{DLOGTIME}$-uniformity, as it is the case for the $\mathbf{TC}^0$ circuits for addition, subtraction, and multiplication (see Barrington, Immerman and Straubing [5]).

On the other hand, it is known that majority gates of polylogarithmically-many bits may be simulated by $\mathbf{DLOGTIME}$-uniform $\mathbf{AC}^0$ circuits (see [34] for a similar construction). A circuit $\mathrm{TH}_k(x_1,\ldots,x_m)$ computing whether at least k of the input bits $x_1,\ldots,x_m$ is recursively built as follows:

$$\mathrm{TH}_k(x_1,\ldots,x_m) := \bigvee_{\substack{i_1+\cdots+i_s\geq k\\ i_j\leq m/s}} \bigwedge_{j=1}^{s} \mathrm{TH}_{i_j}(x_{(j-1)m/s+1},\ldots,x_{jm/s}),$$

where $m = (\log n)^{O(1)}$, and s is suitably chosen so that the size of the circuit is polynomial in n, and the depth is a constant independent of n (the choice $s = (\log n)^{\epsilon}$ works for sufficiently small ϵ). It is not hard to see that these circuits are **DLOGTIME**-uniform (a clever numbering of gates will tell all the required information to the **DLOGTIME** algorithm that computes the direct connection language). The well-known power of $\mathbf{AC}^0$ circuits to do arithmetic on numbers with polylogarithmically-many significant bits follows from Reif's result, and the known algorithms for addition, subtraction and multiplication. However, while addition, subtraction and multiplication of polylogarithmically-long numbers admit **DLOGTIME**-uniform $\mathbf{AC}^0$ such circuits, the known algorithms for division fall short since they only give **DPOLYLOGTIME**-uniform $\mathbf{AC}^0$ circuits. We note that Theorem 3 implies that division of numbers with polylogarithmically-many significant bits is definable in $\mathrm{FO} + \mathrm{LFP}(\Delta_0)$ on the class of finite words with built-in membership relation. We do not know, however, of a direct proof of this fact.

Acknowledgments I am grateful to José L. Balcázar and Phokion Kolaitis for insightful comments, and to Ricard Gavaldà for teaching me about the simulation of $\mathbf{TC}^0$ circuits of polylogarithmically many bits by $\mathbf{AC}^0$ circuits. I am also grateful to Martin Grohe who asked the question that led to Theorem 3.

References

[1] E. Allender and V. Gore. Rudimentary reductions revisited. *Information Processing Letters*, 40:89–95, 1991.

[2] E. Allender and V. Gore. A uniform circuit lower bound for the permanent. *SIAM Journal of Computing*, 23(5):1026–1049, 1994.

[3] A. Atserias and Ph. G. Kolaitis. First-order logic vs. fixed-point logic in finite set theory. In *14th IEEE Symposium on Logic in Computer Science*, pages 275–284, 1999.

[4] D. M. Barrington and N. Immerman. Time, hardware, and uniformity. In *Complexity Theory Retrospective II*, pages 1–22. Springer-Verlag, 1997.

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

[6] J. Barwise. *Admissible Sets and Structures*. Springer-Verlag, 1975.

[7] P. W. Beame, S. A. Cook, and H. J. Hoover. Log depth circuits for division and related problems. *SIAM Journal of Computing*, 15(4):994–1003, 1986.

[8] J. H. Bennett. *On Spectra*. PhD thesis, Princeton University, 1962.

[9] S. R. Buss. The boolean function value problem is in ALOGTIME. In *28th Annual IEEE Symposium on Foundations of Computer Science*, pages 123–131, 1987.

[10] A. K. Chandra, D. C. Kozen, and L. J. Stockmeyer. Alternation. *Journal of the ACM*, 28:114–133, 1981.

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

[12] A. Dawar and L. Hella. The expressive power of finitely many generalized quantifiers. *Information and Computation*, 123:172–184, 1995.

[13] A. Dawar, S. Lindell, and S. Weinstein. First order logic, fixed point logic and linear order. In *Computer Science Logic '95*, volume 1092 of *Lecture Notes in Computer Science*, pages 161–177. Springer-Verlag, 1996.

[14] L. Fortnow. Time-space tradeoffs for satisfiability. In *12th IEEE Conference in Computational Complexity*, pages 52–60, 1997. To appear in *Journal of Computer and System Sciences*.

[15] Y. Gurevich, N. Immerman, and S. Shelah. McColm's conjecture. In *9th IEEE Symposium on Logic in Computer Science*, pages 10–19, 1994.

[16] Y. Gurevich and S. Shelah. Fixed-point extensions of first-order logic. *Annals of Pure and Applied Logic*, 32(3):265–280, 1986.

[17] N. Immerman. Relational queries computable in polynomial time. *Information and Computation*, 68:86–104, 1986.

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

[19] N. Immerman and S. Landau. The complexity of iterated multiplication. *Information and Computation*, 116(1):103–116, 1995.

[20] N. Jones. Context-free languages and rudimentary attributes. *Mathematical Systems Theory*, 3:102–109, 1969.

[21] N. D. Jones. Space-bounded reducibility among combinatorial problems. *Journal of Computer and System Sciences*, 11:68–85, 1975. Corrigendum: *Journal of Computer and System Sciences* 15:241, 1977.

[22] Ph. G. Kolaitis and M. Y. Vardi. Fixpoint logic vs. infinitary logic in finite-model theory. In *7th IEEE Symposium on Logic in Computer Science*, pages 46–57, 1992.

[23] S. Lindell. A purely logical characterization of circuit uniformity. In *7th IEEE Structure in Complexity Theory*, pages 185–192, 1992.

[24] R. J. Lipton and A. Viglas. On the complexity of SAT. In *40th Annual IEEE Symposium on Foundations of Computer Science*, pages 459–464, 1999.

[25] J. A. Makowsky. Invariant definability and P/poly. To appear in Lecture Notes in Computer Science, Proceedings of Computer Science Logic 1998, 1999.

[26] Y. N. Moschovakis. *Elementary Induction on Abstract Structures*. North-Holland, 1974.

[27] V. A. Nepomnjascii. Rudimentary predicates and Turing calculations. *Soviet Math. Dokl.*, 11:1462–1465, 1970.

[28] J. H. Reif. On threshold circuits and polynomial computation. In *2nd IEEE Structure in Complexity Theory*, pages 118–123, 1987.

[29] W. L. Ruzzo. On uniform circuit complexity. *Journal of Computer and System Sciences*, 22:365–383, 1981.

[30] V. Y. Sazonov. On bounded set theory. In *Logic and Scientific Methods*, pages 85–103. Kluwer Academic Publishers, 1997.

[31] M. Sipser. Borel sets and circuit complexity. In *15th Annual ACM Symposium on the Theory of Computing*, pages 61–69, 1983.

[32] R. Smullyan. Theory of formal systems. In *Annals of Mathematics Studies*, volume 47. Princeton University Press, 1961.

[33] L. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3:1–22, 1977.

[34] I. Wegener. *The Complexity of Boolean Functions*, pages 243–247. John Wiley & Sons, 1987.

[35] C. Wrathall. Rudimentary predicates and relative computation. *SIAM Journal of Computing*, 7(2):194–209, 1978.