Journal of Computer and System Sciences 80 (2014) 1152–1162

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

A characterization of definability of second-order generalized quantifiers with applications to non-definability

Juha Kontinen [a,*], Jakub Szymanik [b]

[a] Department of Mathematics and Statistics, University of Helsinki, Finland
[b] Institute for Logic, Language, and Computation, University of Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Received 14 January 2012
Received in revised form 21 February 2013
Accepted 3 February 2014
Available online 4 April 2014

Keywords:
Second-order generalized quantifiers
Definability
Majority quantifier
Second-order logic
Collective quantification

ABSTRACT

We study definability of second-order generalized quantifiers. We show that the question whether a second-order generalized quantifier $\mathcal{Q}_1$ is definable in terms of another quantifier $\mathcal{Q}_2$, the base logic being monadic second-order logic, reduces to the question if a quantifier $\mathcal{Q}_1^\star$ is definable in $\mathrm{FO}(\mathcal{Q}_2^\star, <, +, \times)$ for certain first-order quantifiers $\mathcal{Q}_1^\star$ and $\mathcal{Q}_2^\star$. We use our characterization to show new definability and non-definability results for second-order generalized quantifiers. We also show that the monadic second-order majority quantifier Most^1 is not definable in second-order logic.

1. Introduction

The notion of generalized quantifier goes back to Mostowski [1] and Lindström [2]. Generalized quantifiers were first mainly studied in the framework of model theory. The study of generalized quantifiers extended to the context of finite model theory via applications to descriptive complexity theory. We refer to [3] and [4] for surveys of first-order generalized quantifiers in finite model theory. Generalized quantifiers have been also extensively studied in the formal semantics of natural language (see [5] for a survey).

The study of second-order generalized quantifiers is a relatively new and unexplored area in finite model theory. On the other hand, second-order logic (SO) and its many fragments have been studied extensively starting from Fagin's characterization of NP in terms of existential second-order logic [6]. Second-order generalized quantifiers were first studied in the context of finite structures by Burtschick and Vollmer [7]. Shortly after, Andersson [8] studied the expressive power of families of second-order generalized quantifiers determined by the syntactic types of quantifiers. In [9–11] Kontinen studied definability questions of second-order generalized quantifiers. In the case of first-order quantifiers, definability of a quantifier Q in a logic $\mathcal{L}$ means that the class of structures, used to interpret Q, is axiomatizable in $\mathcal{L}$. In the second-order case, the analogous concept of definability was formulated in [9,10]. In this article, we give a computationally motivated characterization for the notion of definability of second-order generalized quantifiers. Burtschick and Vollmer [7] noticed that second-order generalized quantifiers can be used to logically characterize complexity classes defined in terms of so-called *Leaf Languages*. The leaf languages approach in computational complexity theory, introduced by Bovet, Crescenzi, and Silvestri [12], is a unifying approach to define complexity classes. The central idea behind this approach is to generalize the conditions under which, e.g., a Turing machine or an automaton accepts its input. Many complexity classes can be defined in this context in terms of suitable leaf languages. On the other hand, a complexity

* Corresponding author.

http://dx.doi.org/10.1016/j.jcss.2014.04.007
0022-0000/

class defined in terms of a leaf language B can be under certain conditions characterized logically by a logic of the form:

$$\mathcal{Q}_B\mathrm{FO},$$

where $\mathcal{Q}_B$ is a second-order generalized quantifier corresponding to the language B. In the context of leaf languages, polynomial time non-deterministic Turing machines can be sometimes replaced by non-deterministic finite automata (so-called finite leaf automata) without a significant decrease in complexity [13]. Galota and Vollmer [14] showed that complexity classes defined by finite leaf automata can be logically characterized in terms of monadic second-order generalized quantifiers. This result nicely extends the well known [15–17] characterization of regular languages in terms of monadic second-order logic (MSO).

The definability theory of second-order generalized quantifiers has some similarities and differences compared to that of first-order generalized quantifiers. For example, it was observed in [9] that the binary second-order existential quantifier cannot be defined in terms of any monadic second-order generalized quantifiers. This result is in contrast with the fact (a corollary of a result of Andersson [8]) that all classes of finite first-order structures are already definable in terms of monadic second-order generalized quantifiers.

In this paper we prove a general result characterizing the question when a quantifier $\mathcal{Q}$ is definable in $\mathrm{MSO}(\mathcal{Q}', +)$, where $+$ denotes the built-in addition relation. We assume the built-in addition in order to unleash the expressive power embodied by MSO. Recall that, while MSO corresponds to regular languages over strings, $\mathrm{MSO}(+)$ corresponds to the linear fragment of the polynomial hierarchy (LINH) on strings [18]. Some of our results can be generalized to the case where the base logic is full second-order logic instead of $\mathrm{MSO}(+)$. Our characterization is based on the idea connecting oracle separation results with lower bound results for small constant depth circuits, see e.g., [19–23]. We show that a second-order generalized quantifier $\mathcal{Q}_1$ is definable in the logic $\mathrm{MSO}(\mathcal{Q}_2, +)$ iff for certain first-order encodings $\mathcal{Q}_i^\star$ of $\mathcal{Q}_i$, $\mathcal{Q}_1^\star$ is definable in $\mathrm{FO}(\mathcal{Q}_2^\star, +, \times)$. It is worth noting that the latter condition implies that $\mathcal{Q}_1^\star$ is AC^0 (Turing) reducible to $\mathcal{Q}_2^\star$. We use our characterization to show new definability and non-definability results for second-order generalized quantifiers. In particular, we show that the monadic second-order majority quantifier Most[1] is not definable in second-order logic. This answers the question left open in [24] (see also [25]), where second-order generalized quantifiers were used to model collective quantification in natural language, for example:

1. Most girls gathered.
2. All soldiers surrounded the Alamo.

The common strategy in formalizing collective quantification has been to define the meanings of collective determiners, quantifying over collections, using certain type-shifting operations. These operations, i.e., lifts, define the collective interpretations of determiners systematically from the standard meanings of quantifiers (see, e.g., [26,27]). In [24] we show that all these lifts are definable in second-order logic. In this paper we prove that some collective quantifiers (second-order generalized quantifiers) are not definable in second-order logic. Therefore, there is no second-order definable lift expressing their collective meaning. This is clearly a restriction of the type-shifting approach. One possible alternative would be to use second-order generalized quantifiers in the study of collective semantics, as we already proposed in [24]. However, as it follows from this paper the computational complexity of such approach is excessive and hence it may not be a plausible model of collective quantification in natural language (see [28–30] for a discussion of computational restrictions in natural language semantics). Hence, it may be wise to turn in the direction of another well-known way of studying collective quantification in natural language, the many-sorted (algebraic) tradition (see [31]). Another linguistic interpretation of our results might be that computational complexity restricts the expressive power of everyday language (see [32]). Namely, even though natural language can in principle realize collective quantifiers non-definable in second-order logic, its everyday fragment does not contain such constructions due to their high complexity.

2. Preliminaries

In this article all structures are assumed to be finite. The universe of a structure $\mathfrak{A}$ is denoted by A. Without loss of generality, we may assume that A is always of the form $\{0, \dots, m\}$ for some $m \in \mathbb{N}$. For a logic $\mathcal{L}$, the set of τ-formulas of $\mathcal{L}$ is denoted by $\mathcal{L}[\tau]$. If ϕ is a τ-sentence, then the class of τ-models of ϕ is denoted by $\mathrm{Mod}(\phi)$. A class K of τ-models is said to be axiomatizable in a logic $\mathcal{L}$, if $K = \mathrm{Mod}(\phi)$ for some sentence $\phi \in \mathcal{L}[\tau]$. For logics $\mathcal{L}$ and $\mathcal{L}'$, we write $\mathcal{L} \leqslant \mathcal{L}'$, if for every τ and every sentence $\phi \in \mathcal{L}[\tau]$ there is a sentence $\psi \in \mathcal{L}'[\tau]$ such that $\mathrm{Mod}(\phi) = \mathrm{Mod}(\psi)$. The set of natural numbers is denoted by $\mathbb{N}$ and $\mathbb{N}^*$ denotes the set $\mathbb{N} \setminus \{0\}$.

Sometimes we assume that our structures (and logics) are equipped with auxiliary built-in relations. In addition to the built-in ordering $<$, which is interpreted naturally, we also use the ternary relations $+$ and $\times$. The relations $+$ and $\times$ are defined as

$$+(i, j, k) \quad \Leftrightarrow \quad i + j = k,$$
$$\times(i, j, k) \quad \Leftrightarrow \quad i \times j = k.$$

The relation BIT is a further important relation which is defined by: $\mathrm{BIT}(a, j)$ holds iff the bit of order 2^j is 1 in the binary representation $\mathrm{bin}(a)$ of a. The presence of built-in relations is signaled, e.g., by the notation $\mathrm{FO}(<)$. It is well known that $\mathrm{FO}(<, +, \times) \equiv \mathrm{FO}(<, \mathrm{BIT})$ (see [33]). Note that $<$ is easily definable in $\mathrm{FO}(+)$ and hence, in the presence of $+$, we sometimes do not mention $<$ explicitly.

We assume that the reader is familiar with the basics of computational complexity theory. Below, we recall certain results from descriptive complexity theory. It is instructive to note that many of the logics considered in this article correspond to interesting complexity classes. We mention first the logic $\mathrm{FO}(<, +, \times)$ which corresponds exactly to the so-called logarithmic hierarchy (LH). This class is the logarithmic analogue of the polynomial hierarchy (PH), corresponding to SO [34], defined in terms of alternating Turing machines (ATM) running in polynomial time with $O(1)$ alternations. In between LH and PH we have the linear hierarchy (LINH) corresponding to the logic $\mathrm{MSO}(+)$ over strings [18].

In this article also majority quantifiers are discussed and studied. It is well-known that majority quantifiers can be used to logically characterize counting computations. The following counting hierarchies are relevant for this article: the logarithmic counting hierarchy (LCH), the linear counting hierarchy (LINCH), and the (polynomial) counting hierarchy (CH) all of which can be defined, with analogous resource bounds as LH, LINH, and PH, in terms of so-called Threshold Turing machines [35]. On the logical side, majority quantifiers (defined in Section 2.1) can be used to provide logical counterparts for these classes: $\mathrm{FO}(\mathrm{M}, +, \times) \equiv \mathrm{LCH}$ [36], $\mathrm{FO}(\mathrm{Most}^1, <) \equiv \mathrm{LINCH}$ (over strings) [37], and $\mathrm{FO}(\mathrm{Most}^k)_{k\in\mathbb{N}^*} \equiv \mathrm{CH}$ [38]. Furthermore, in circuit complexity, it is known that LH corresponds exactly to DLOGTIME-uniform AC^0 and LCH to DLOGTIME-uniform TC^0 [36]. Also, DLOGTIME-uniform $\mathrm{AC}^0[p]$ (AC^0 with unbounded fan-in MOD_p gates) corresponds on the logical side to $\mathrm{FO}(\mathrm{D}_k, +, \times)$ [36].

2.1. Generalized quantifiers

In this section we briefly recall some basics of generalized quantifiers.

Let $\tau = \{P_1, \ldots, P_r\}$ be a relational vocabulary, where P_i is l_i-ary for $1 \leqslant i \leqslant r$, and Q a class of τ-structures closed under isomorphisms. The class Q gives rise to a generalized quantifier which we also denote by Q. The tuple $s = (l_1, \ldots, l_r)$ is the *type* of the quantifier Q.

Definition 2.1. The extension $\mathrm{FO}(Q)$ of first-order logic by a quantifier Q is defined as follows:

1. The formula formation rules of FO are extended by the rule: if for $1 \leqslant i \leqslant r$, $\phi_i(\bar{x}_i)$ is a formula and $\bar{x}_i$ is an l_i-tuple of pairwise distinct variables then $Q\bar{x}_1, \ldots, \bar{x}_r(\phi_1(\bar{x}_1), \ldots, \phi_r(\bar{x}_r))$ is a formula.
2. The satisfaction relation of FO is extended by the rule:
$$\mathfrak{A} \models Q\bar{x}_1, \ldots, \bar{x}_r\big(\phi_1(\bar{x}_1), \ldots, \phi_r(\bar{x}_r)\big) \quad \text{iff } \big(A, \phi_1^{\mathfrak{A}}, \ldots, \phi_r^{\mathfrak{A}}\big) \in Q,$$
where $\phi_i^{\mathfrak{A}} = \{\bar{a} \in A^{l_i} \mid \mathfrak{A} \models \phi_i(\bar{a})\}$.

We say that a quantifier Q is definable in a logic $\mathcal{L}$ if the class Q is axiomatizable in $\mathcal{L}$. Note that Q is trivially definable in $\mathrm{FO}(Q)$. If $\mathcal{L}$ has the substitution property and is closed under FO-operations, then definability of Q in $\mathcal{L}$ implies that $\mathrm{FO}(Q) \leqslant \mathcal{L}$. So, among such logics, $\mathrm{FO}(Q)$ is the minimal logic in which Q is definable.

Example 2.2. The following quantifiers will be discussed in the sections to come. Suppose $S \subseteq \mathbb{N}$ and $k \in \mathbb{N}$.

$$\begin{aligned}
\exists &= \big\{(A, P) \bigm| P \subseteq A \text{ and } P \neq \emptyset\big\} \\
\mathrm{M} &= \big\{(A, P) \bigm| P \subseteq A \text{ and } |P| > |A|/2\big\} \\
Q_S &= \big\{(A, P) \bigm| P \subseteq A \text{ and } |P| \in S\big\} \\
\mathrm{I} &= \big\{(A, P_1, P_2) \bigm| P_i \subseteq A \text{ and } |P_1| = |P_2|\big\}
\end{aligned}$$

If S is of the form $\{kn \mid n \in \mathbb{N}\}$ for some $k \in \mathbb{N}$, we denote Q_S by D_k.

We will also refer to the *vectorizations* of the quantifiers D_k and M later. The nth vectorization of D_k is the following quantifier

$$\mathrm{D}_k^n = \big\{(A, P) \bigm| P \subseteq A^n \text{ and } |P| = 0 \bmod k\big\},$$

and the nth vectorization of M is

$$\mathrm{M}^n = \big\{(A, P) \bigm| P \subseteq A^n \text{ and } |P| > \big|A^n\big|/2\big\}.$$

Let us then turn to second-order generalized quantifiers. Let $t = (s_1, \ldots, s_w)$, where $s_i = (l_1^i, \ldots, l_{r_i}^i)$ is a tuple of positive integers for $1 \leqslant i \leqslant w$. A second-order structure of type t is a structure of the form $(A, P_1, \ldots, P_w)$, where $P_i \subseteq \mathcal{P}(A^{l_1^i}) \times \cdots \times \mathcal{P}(A^{l_{r_i}^i})$.

Definition 2.3. A second-order generalized quantifier $\mathcal{Q}$ of type t is a class of structures of type t such that $\mathcal{Q}$ is closed under isomorphisms.

A quantifier $\mathcal{Q}$ is *monadic* if $l^i_j = 1$ for all $1 \leqslant j \leqslant r_i$ and $1 \leqslant i \leqslant w$. Let us look at some examples of second-order generalized quantifiers.

Example 2.4. Suppose $S \subseteq \mathbb{N}$ and $k \in \mathbb{N}$.

$$\begin{aligned}
\exists^2_k &= \bigl\{(A, P) \bigm| P \subseteq \mathcal{P}\bigl(A^k\bigr) \text{ and } P \neq \emptyset\bigr\}\\
\text{Even} &= \bigl\{(A, P) \bigm| P \subseteq \mathcal{P}(A) \text{ and } |P| \text{ is even}\bigr\}\\
\text{Even}' &= \bigl\{(A, P) \bigm| P \subseteq \mathcal{P}(A) \text{ and } \forall X \in P\ (|X| \text{ is even})\bigr\}\\
\text{Most}^k &= \bigl\{(A, P) \bigm| P \subseteq \mathcal{P}\bigl(A^k\bigr) \text{ and } |P| > 2^{|A|^k-1}\bigr\}\\
\mathrm{I}^2 &= \bigl\{(A, P_1, P_2) \bigm| P_i \subseteq \mathcal{P}(A) \text{ and } |P_1| = |P_2|\bigr\}\\
\mathcal{Q}_S &= \bigl\{(A, P) \bigm| P \subseteq \mathcal{P}(A) \text{ and } |P| \in S\bigr\}
\end{aligned}$$

Analogously to the first-order case, if S is of the form $\{kn \mid n \in \mathbb{N}\}$ for some $k \in \mathbb{N}$, we denote $\mathcal{Q}_S$ by $\mathcal{D}_k$.

The first example is the familiar k-ary second-order existential quantifier. The quantifier Even says that a formula holds for an even number of subsets of the universe. On the other hand, the quantifier Even$'$ says that all the subsets satisfying a formula have an even cardinality. The quantifier Most^k is the k-ary second-order version of M expressing that a formula holds for more than half of the k-ary relations.

Definition 2.5. The extension FO($\mathcal{Q}$) of FO by a quantifier $\mathcal{Q}$ is defined as follows:

1. The formula formation rules of FO are extended by the rule: if for $1 \leqslant i \leqslant w$, $\phi_i(\overline{X}_i)$ is a formula and $\overline{X}_i = (X_{1,i}, \ldots, X_{r_i,i})$ is a tuple of pairwise distinct predicate variables such that the arity of $X_{j,i}$ is l^i_j for $1 \leqslant j \leqslant r_i$, then
$$\mathcal{Q}\overline{X}_1, \ldots, \overline{X}_w\bigl(\phi_1(\overline{X}_1), \ldots, \phi_w(\overline{X}_w)\bigr)$$
is a formula.
2. Satisfaction relation of FO is extended by the rule:
$$\mathfrak{A} \models \mathcal{Q}\overline{X}_1, \ldots, \overline{X}_w(\phi_1, \ldots, \phi_w) \quad \text{iff } \bigl(A, \phi_1^{\mathfrak{A}}, \ldots, \phi_w^{\mathfrak{A}}\bigr) \in \mathcal{Q},$$
where $\phi_i^{\mathfrak{A}} = \{\overline{R} \in \mathcal{P}(A^{l^i_1}) \times \cdots \times \mathcal{P}(A^{l^i_{r_i}}) \mid \mathfrak{A} \models \phi_i(\overline{R})\}$.

2.2. Definability

Recall that a first-order generalized quantifier Q is definable in a logic $\mathcal{L}$ if the class Q is axiomatizable in $\mathcal{L}$. This condition can be reformulated as follows assuming $\mathcal{L}$ has the substitution property:

Proposition 2.6. *A first-order quantifier Q is definable in a logic $\mathcal{L}$ if and only if $\mathcal{L} \equiv \mathcal{L}(Q)$.*

How do we formalize definability for second-order quantifiers? Intuitively, e.g., the monadic second-order existential quantifier $\exists^2_1$ is definable in a logic $\mathcal{L}$ if there is a uniform way to express

$$\exists^2_1 X \psi(X)$$

for any formula $\psi(X)$ in the logic $\mathcal{L}$. Over a model $\mathfrak{A}$, $\psi(X)$ defines a collection of subsets

$$\bigl\{C \subseteq A \mid \mathfrak{A} \models \psi(C)\bigr\},$$

so the problem is to find a way to express the non-emptiness of this collection in a way which does not depend on the particular formula $\psi(X)$. This was formalized in [10] using second-order relations.

Definition 2.7. Let $\mathcal{L}$ be a logic, $t = (s_1, \ldots, s_w)$ a second-order type, and let $\mathcal{G}_1, \ldots, \mathcal{G}_w$ be first-order quantifier symbols of types $s_1, \ldots, s_w$.

1. The logic $\mathcal{L}(\mathcal{G}_1, \ldots, \mathcal{G}_w)$ is obtained by extending the syntax of $\mathcal{L}$ in terms of the quantifiers $\mathcal{G}_1, \ldots, \mathcal{G}_w$.

2. The models of $\mathcal{L}(\mathcal{G}_1, \ldots, \mathcal{G}_w)$ are of the form $\mathcal{A} = (\mathfrak{A}, G_1, \ldots, G_w)$, where $\mathfrak{A}$ is a first-order model and

$$G_i \subseteq \mathcal{P}\left(A^{l^i_1}\right) \times \cdots \times \mathcal{P}\left(A^{l^i_{r_i}}\right).$$

3. The quantifiers $\mathcal{G}_i$ are interpreted using the relations G_i:

$$\mathcal{A} \models \mathcal{G}_i \bar{x}_1, \ldots, \bar{x}_{r_i}\left(\phi_1(\bar{x}_1), \ldots, \phi_{r_i}(\bar{x}_{r_i})\right)$$

iff $(\phi_1^{\mathcal{A}}, \ldots, \phi_{r_i}^{\mathcal{A}}) \in G_i$.

Note that if $\phi \in \mathcal{L}(\mathcal{G}_1, \ldots, \mathcal{G}_w)$ is a sentence of vocabulary $\tau = \emptyset$. Then

$$\mathrm{Mod}(\phi) = \left\{(A, G_1, \ldots, G_w) \;\middle|\; (A, G_1, \ldots, G_w) \models \phi\right\}$$

corresponds to a second-order generalized quantifier of type t. This observation can be used to formalize definability of second-order generalized quantifiers. Below, we assume that $\mathcal{L}$ is closed under substitution.

Definition 2.8. Let $\mathcal{Q}$ be a quantifier of type t. The quantifier $\mathcal{Q}$ is definable in a logic $\mathcal{L}$ if there is $\phi \in \mathcal{L}(\mathcal{G}_1, \ldots, \mathcal{G}_w)$ of vocabulary $\sigma = \emptyset$ such that for any t-structure $(A, G_1, \ldots, G_w)$,

$$(A, G_1, \ldots, G_w) \models \phi \quad \Leftrightarrow \quad (A, G_1, \ldots, G_w) \in \mathcal{Q}.$$

The following was shown in [10]:

Theorem 2.9. *If $\mathcal{Q}$ is definable in $\mathcal{L}$ then $\mathcal{L} \equiv \mathcal{L}(\mathcal{Q})$.*

The converse of Theorem 2.9 does not hold:

Theorem 2.10. *(See [10].) There is a quantifier $\mathcal{Q}$ of type $((1))$ which is not definable in* FO *and satisfies* $\mathrm{FO} \equiv \mathrm{FO}(\mathcal{Q})$.

Definability questions of second-order quantifiers have been studied in [10,11,38]. We recall the following results.

Theorem 2.11. *(See [11].) Let t be type and $\mathcal{B}_t$ the collection of all second-order quantifiers of types less than t. Then there is a quantifier $\mathcal{Q}$ of type t such that $\mathcal{Q}$ is not definable in* $\mathrm{SO}(\mathcal{B}_t)$.

Theorem 2.11 is proved with respect to a natural ordering of the types of second-order generalized quantifiers. Theorem 2.11 is existential in nature and does not give us a concrete non-definable quantifier. It was observed in [9] that it is not so difficult to find concrete quantifiers which cannot be defined using any monadic quantifiers.

Denote by $\mathbb{Q}$ the collection of all monadic second-order generalized quantifiers.

Theorem 2.12. *(See [9].) The quantifier $\exists^2_2$ is not definable in* $\mathrm{FO}(\mathbb{Q})$.

It is worth noting that the logic $\mathrm{FO}(\mathbb{Q})$ is capable of defining all classes of first-order structures (cf. Theorem 6.2 in [8]). Finally, we recall the following result about second-order majority quantifiers:

Theorem 2.13. *(See [38].) The quantifier $\exists^2_k$ is definable in* $\mathrm{FO}(\mathrm{Most}^k)$.

It interesting to note that definability of Most^1 in the logic SO would imply that $\mathrm{PH} \equiv \mathrm{CH}$ in computational complexity. This observation was discussed in [24,25]. In this paper we show that the quantifier Most^1 is not definable in SO, but, analogously to Theorem 2.10, this non-definability result does not imply that $\mathrm{PH} \subsetneq \mathrm{CH}$.

3. Characterizing definability

The computational analogue of a first-order generalized quantifier is the notion of an oracle (see [33]). Let Q be a quantifier of vocabulary τ and $\mathcal{L}$ a logic. The idea is that in $\mathcal{L}(Q)$ we can query "without a cost" if a definable τ-structure $\mathfrak{A}$ is a member of the class Q. Recall that a second-order generalized quantifier $\mathcal{Q}$ of type $((1))$ is definable, e.g., in SO if there is a sentence $\phi \in \mathrm{SO}(\mathcal{G})$ such that for all second-order structures (A, G):

$$(A, G) \models \phi \quad \Leftrightarrow \quad (A, G) \in \mathcal{Q}. \tag{1}$$

It is not immediately clear how to view this notion in computational terms. The set G corresponds to a local first-order quantifier and, if we treat G as an oracle, then in (1) we are in fact trying to define a property oracles. One way to proceed is to formalize definability of a quantifier $\mathcal{Q}$ in terms of oracle Turing machines that treat (a suitable initial segment) the oracle as part of the input. However, in this article we do not follow that idea as there is a more familiar route to take. An important observation here is that the set G can be of exponential size compared to the domain A. This observation can be used to show that SO-definability of $\mathcal{Q}$ reduces to logarithmic time definability.

Our result is a logical version of the results connecting oracle separation results with lower bound results for small constant depth circuits, see e.g., [19–23]. For example, in [22], Torán studied oracle separations in the counting hierarchy and noticed that there is essentially no difference between an oracle Turing machine writing an oracle query on its query tape and a logarithmic time Turing machine writing an address on its random access tape. He used this analogy to show that an oracle separation result for classes in the polynomial counting hierarchy implies a real separation for the corresponding classes in the logarithmic counting hierarchy LINCH (equivalently in DLOGTIME-uniform TC^0).

We use a logical version of this idea: we show that SO and the relation G in (1) can be replaced by FO and a unary relation P by passing from A to a domain of cardinality $2^{|A|}$.

In this section we mainly restrict attention to monadic second-order generalized quantifiers. We interpret definability of second-order quantifiers in MSO(+) in the natural way: for example, a second-order quantifier $\mathcal{Q}$ of type ((1)) is definable in MSO(+) if there is $\phi \in \mathrm{MSO}(\mathcal{G}, +)$ such that for all structures $(A, +, G)$: $(A, +, G) \models \phi \Leftrightarrow (A, G) \in \mathcal{Q}$. In particular, Theorem 2.9 can be proved analogously in this setting.

Next we define a first-order encoding of a second-order structure of type t, for a monadic t. We use the fact that there is a one-to-one correspondence between integers $m \in B = \{0, \ldots, 2^n - 1\}$ and subsets of $A = \{0, \ldots, n-1\}$ seen as length-n binary numbers. Therefore, relations of A can be encoded in terms of tuples of elements of B and, further, sets of relations of A by relations of B.

Definition 3.1. Let $t = (s_1, \ldots, s_w)$ be a type where $s_i = (1, \ldots, 1)$ is of length r_i for $1 \leqslant i \leqslant w$. Let $\mathfrak{A} = (A, G_1, \ldots, G_w)$ be a t-structure where $A = \{0, \ldots, n-1\}$ and $G_i \subseteq \mathcal{P}(A) \times \cdots \times \mathcal{P}(A)$. Denote by $\hat{\mathfrak{A}} = (B, P_1, \ldots, P_w)$ the following first-order structure of vocabulary $\tau = \{P_1, \ldots, P_w\}$, where P_i is an r_i-ary predicate, and

1. $B = \{0, \ldots, 2^n - 1\}$,
2. $P_i = \{(j_1, \ldots, j_{r_i}) \in B^{r_i} \mid (J_1, \ldots, J_{r_i}) \in G_i\}$, where, for $1 \leqslant k \leqslant r_i$, the length-$n$ binary representation of j_k is given by $s_0 \cdots s_{n-1}$, and $s_l = 1 \Leftrightarrow l \in J_k$.

For a quantifier $\mathcal{Q}$ of type t, we denote by $\mathcal{Q}^\star$ the first-order quantifier of vocabulary τ defined by

$$\mathcal{Q}^\star := \{\hat{\mathfrak{A}} \colon \mathfrak{A} \in \mathcal{Q}\}.$$

It is easy to see that the quantifier $\mathcal{Q}^\star$ has only structures in cardinalities of the form 2^n and that $|G_i| = |P_i|$ for $1 \leqslant i \leqslant w$. Note also that the quantifier $\mathcal{Q}^\star$ encoding $\mathcal{Q}$ may depend on the ordering of the domain B and hence does not strictly speaking correspond to a Lindström quantifier of vocabulary τ but a τ-quantifier with build-in arithmetic relations (quantifiers defined and studied in [39]). On the other hand, for the numerical quantifiers $\mathcal{Q}$ discussed in the rest of this section, the first-order encodings $\mathcal{Q}^\star$ are obviously order invariant and hence correspond to Lindström quantifiers of vocabulary τ. We are now ready for the main result of this article.

Theorem 3.2. *Let $\mathcal{Q}_1$ and $\mathcal{Q}_2$ be monadic quantifiers. Then $\mathcal{Q}_1$ is definable in* $\mathrm{MSO}(\mathcal{Q}_2, +)$ *if and only if $\mathcal{Q}_1^\star$ is definable in* $\mathrm{FO}(\mathcal{Q}_2^\star, +, \times)$.

Proof. To simplify notation, we assume that the type of $\mathcal{Q}_1$ and $\mathcal{Q}_2$ is ((1, 1)) and ((1), (1)), respectively.

Let us first assume that $\mathcal{Q}_1$ is definable in the logic $\mathrm{MSO}(\mathcal{Q}_2, +)$. Then there is a sentence $\phi \in \mathrm{MSO}(\mathcal{Q}_2, \mathcal{G}, +)$ such that for all structures $(A, +, G)$

$$(A, +, G) \models \phi \quad \Leftrightarrow \quad (A, G) \in \mathcal{Q}_1.$$

We shall next show that there is a sentence $\phi^* \in \mathrm{FO}(\mathcal{Q}_2^\star +, \times)[\{P\}]$, where P is binary, such that for all structures $\mathfrak{A} = (A, G)$:

$$(A, +, G) \models \phi \quad \Leftrightarrow \quad (B, P, <, +, \times) \models \phi^*, \tag{2}$$

where $(B, P) = \hat{\mathfrak{A}}$ (see Definition 3.1). We define ϕ^* via the following translation, where x_i and Y_i denote the first-order and the unary predicate variables appearing in the formulas of $\mathrm{MSO}(\mathcal{Q}_2, \mathcal{G}, +)$.

$$\begin{aligned}
x_i = x_j &\rightsquigarrow x_i = x_j \\
x_i + x_j = x_k &\rightsquigarrow x_i + x_j = x_k \\
Y_i(x_j) &\rightsquigarrow \mathrm{BIT}\big(y_i, n-(x_j+1)\big) \\
\mathcal{G}x_i, x_j\big(\psi_1(x_i), \psi_2(x_j)\big) &\rightsquigarrow \exists z_1 \exists z_2 \Big(P(z_1, z_2) \wedge \bigwedge_{1 \leqslant i \leqslant 2} \forall (w < n)\big(\psi_i^*(w) \leftrightarrow \mathrm{BIT}\big(z_i, n-(w+1)\big)\big)\Big) \\
\psi \wedge \theta &\rightsquigarrow \psi^* \wedge \theta^* \\
\neg\psi &\rightsquigarrow \neg\psi^* \\
\exists x_i \psi &\rightsquigarrow \exists x_i \big(x_i < n \wedge \psi^*(x_i)\big) \\
\exists Y_i \psi &\rightsquigarrow \exists y_i \psi^* \\
\mathcal{Q}_2 Y_i, Y_j\big(\psi(Y_i), \theta(Y_j)\big) &\rightsquigarrow \mathcal{Q}_2^\star y_i, y_j\big(\psi^*(y_i), \theta^*(y_j)\big)
\end{aligned}$$

It is now straightforward to show that for all formulas $\psi \in \mathrm{MSO}(\mathcal{Q}_2, \mathcal{G}, +)$, structures (A, G), and assignments s

$$(A, +, G) \models_s \psi \quad \Leftrightarrow \quad (B, P, <, +, \times) \models_{s^*} \psi^*,$$

where the assignment s^* is defined such that $s^*(x_i) = s(x_i)$ for all first-order variables x_i, and, if $s(Y_i) = D \subseteq \{0, \ldots, n-1\}$, then $s^*(y_i)$ is the unique $d < 2^n$ whose binary representation is given by $s_0 \cdots s_{n-1}$ where $s_j = 1 \iff j \in D$.

In the formula translation, we use the predicate BIT, which is FO$(+, \times)$-definable, to recover the set D from the integer d. By the above translation, the sentence

$$\exists n\big(|B| = 2^n \wedge \phi^*\big)$$

of the logic FO$(\mathcal{Q}_2^\star, +, \times)$ now defines the quantifier $\mathcal{Q}_1^\star$.

Let us then show the converse implication. Assume that $\phi \in \mathrm{FO}(\mathcal{Q}_2^\star, +, \times)$ defines the quantifier $\mathcal{Q}_1^\star$. The idea is now to translate $\phi \in \mathrm{FO}(\mathcal{Q}_2^\star, +, \times)$ to $\phi' \in \mathrm{MSO}(\mathcal{Q}_2, \mathcal{G}, +)$ such that for all $\mathfrak{A} = (A, G)$:

$$(A, +, G) \models \phi' \quad \Leftrightarrow \quad (B, P, <, +, \times) \models \phi. \tag{3}$$

Analogously to the first translation, we encode integers in the domain $B = \{0, \ldots, 2^n - 1\}$ in terms of subsets $X \subseteq \{0, \ldots, n-1\}$. We use the following formulas $X = Y$, $X < Y$, $X + Y = Z$, and $X \times Y = Z$ expressing arithmetic operations on binary numbers. The first three formulas are FO$(+)$-expressible, and the fourth is expressible in the logic FO$(\mathrm{M}, +, \times) \leqslant$ MSO$(+)$ [40]. The translation $\phi \rightsquigarrow \phi'$ is now defined as follows.

$$\begin{aligned}
P(x_i, x_j) &\rightsquigarrow \mathcal{G}z_1, z_2\big(X_i(z_1), X_j(z_2)\big) \\
x_i = x_j &\rightsquigarrow X_i = X_j \\
x_i < x_j &\rightsquigarrow X_i < X_j \\
x_i + x_j = x_k &\rightsquigarrow X_i + X_j = X_k \\
x_i \times x_j = x_k &\rightsquigarrow X_i \times X_j = X_k \\
\psi \wedge \phi &\rightsquigarrow \psi' \wedge \phi' \\
\neg\psi &\rightsquigarrow \neg\psi' \\
\exists x_i \psi(x_i) &\rightsquigarrow \exists X_i \psi'(X_i) \\
Q_2^\star x_i, x_j\big(\psi(x_i), \theta(x_j)\big) &\rightsquigarrow \mathcal{Q}_2 X_i, X_j\big(\psi'(X_i), \theta'(X_j)\big)
\end{aligned}$$

It is straightforward to show that this translation works as intended. In particular, it follows that the sentence $\phi' \in$ MSO$(\mathcal{Q}_2, \mathcal{G}, +)$ now defines the quantifier $\mathcal{Q}_1$. $\square$

Let us then discuss some corollaries of Theorem 3.2. We need the following definition.

Definition 3.3. Let $t = (s_1, \ldots, s_w)$ and τ be as in Definition 3.1. Let $\mathcal{Q}$ be a quantifier of type t. The quantifier $\mathcal{Q}$ is *numerical* if there is a relation $T \subseteq \mathbb{N}^w$ such that for all t-structures $(A, P_1, \ldots, P_w)$

$$(A, P_1, \ldots, P_w) \in \mathcal{Q} \quad \Leftrightarrow \quad \big(|P_1|, \ldots, |P_w|\big) \in T.$$

We denote $\mathcal{Q}$ by $\mathcal{Q}_T$ and by Q_T the first-order numerical quantifier (defined analogously) of vocabulary τ.

It is easy to see that, for a numerical $\mathcal{Q}_T$, the quantifier $\mathcal{Q}_T^\star$ (see Definition 3.1) is just the restriction of the corresponding first-order quantifier Q_T to the cardinalities 2^n:

$$\mathcal{Q}_T^\star = \big\{(A, P_1, \ldots, P_w) \in Q_T\colon |A| = 2^n \text{ for some } n \in \mathbb{N}\big\}.$$

This observation allows us to show the following:

Theorem 3.4. *Let $\mathcal{Q}_T$ be a numerical quantifier and $k \in \mathbb{N}$. Then*

1. *$\mathcal{Q}_T$ is definable in* MSO(+) *iff Q_T is definable in* FO$(+, \times)$.
2. *$\mathcal{Q}_T$ is definable in* MSO$(\mathcal{D}_k, +)$ *iff Q_T is definable in* FO$(\mathrm{D}_k, +, \times)$.
3. *$\mathcal{Q}_T$ is definable in* MSO$(\mathrm{Most}^1, +)$ *iff Q_T is definable in* FO$(\mathrm{M}, +, \times)$.

Proof. The proof is based on the fact that each of the logics FO$(<, +, \times)$, FO$(\mathrm{D}_k, +, \times)$, and FO$(\mathrm{M}, +, \times)$ is closed under logical reductions. Suppose that $\mathcal{Q}_T$ is of type $t = (s_1, \ldots, s_w)$ and let τ denote the vocabulary of the corresponding first-order quantifier Q_T (see Definition 3.1).

Let us consider claim 2. By Theorem 3.2 it suffices to show that the following are equivalent:

(a) $\mathcal{Q}_T^\star$ is definable in FO$(\mathcal{D}_k^\star, +, \times)$
(b) Q_T is definable in FO$(\mathrm{D}_k, +, \times)$

Recall that the quantifiers $\mathcal{Q}_T^\star$ and $\mathcal{D}_k^\star$ are the restrictions of the quantifiers Q_T and D_k to cardinalities of the form 2^n, respectively. Let us first note that (a) is equivalent with

(c) $\mathcal{Q}_T^\star$ is definable in FO$(\mathrm{D}_k, +, \times)$.

First of all, since $\mathcal{D}_k^\star$ is easily definable in FO$(\mathrm{D}_k, +, \times)$ using the FO$(+, \times)$-expressible predicate $x = 2^y$, it follows that $(a) \Rightarrow (c)$. Assume then that (c) holds and let $\phi \in$ FO$(\mathrm{D}_k, +, \times)$ define $\mathcal{Q}_T^\star$. Define a sentence ψ as follows:

$$\psi := \exists n\big(|A| = 2^n \wedge \phi\big(\mathrm{D}_k/\mathcal{D}_k^\star\big)\big).$$

Since the quantifier $\mathcal{Q}_T^\star$ contains structures only in cardinalities of the form 2^n it is easy to see that $\psi \in$ FO$(\mathcal{D}_k^\star, +, \times)$ also defines $\mathcal{Q}_T^\star$.

It now suffices to show that (b) and (c) are equivalent. Note first that $(b) \Rightarrow (c)$ can be easily proved using the predicate $x = 2^y$. We will show $(c) \Rightarrow (b)$. Here we use the fact that the logic FO$(\mathrm{D}_k, +, \times)$ is closed under logical reductions. We will define Q_T (over all cardinalities) with the help of the quantifier $\mathcal{Q}_T^\star$. Let $\mathfrak{A}$ be a structure. If $|A| = 2^n$ for some $n \in \mathbb{N}$, then $\mathfrak{A} \in Q_T$ can be expressed in terms of the quantifier $\mathcal{Q}_T^\star$. Note that even if $|A|$ is not a power of two, it holds that the least m such that $|A| \leqslant 2^m$ satisfies $2^m \leqslant 2|A|$.

We will now sketch how the quantifier Q_T can be defined in terms of $\mathcal{Q}_T^\star$. Assume $\phi \in$ FO$(\mathrm{D}_k, +, \times)$ is a sentence defining $\mathcal{Q}_T^\star$. Let $\mathfrak{A} = (A, P_1, \ldots, P_w)$ be a τ-structure, where $A = \{0, \ldots, n-1\}$. We use the following facts:

1. There is a FO$(<, +, \times)$-definable query F that maps $\mathfrak{A}$ to the structure $F(\mathfrak{A})$ which is isomorphic to

$$\big(\{0, \ldots, 2^m - 1\}, P_1, \ldots, P_w, <, +, \times\big),$$

   where 2^m is the least power of two satisfying $n \leqslant 2^m$.
2. There is a sentence $\psi \in$ FO$(\mathrm{D}_k, +, \times)$ such that for all $\mathfrak{A}$:

$$\mathfrak{A} \models \psi \quad \Leftrightarrow \quad F(\mathfrak{A}) \models \phi.$$

Since Q_T is numerical, the sentence ψ now defines Q_T. The query F is easily definable in FO$(<, +, \times)$; the domain of $F(\mathfrak{A})$ is defined as $\{(i, j) \in A^2 \mid in + j < 2^m\}$ (see [33] for more on first-order queries).

The sentence ψ is constructed inductively (see e.g., Section 3.2 [33]) using, in particular, the fact that the second vectorization D_k^2 of D_k can be expressed in FO$(\mathrm{D}_k, +, \times)$.

The claims 1 are 3 are proved analogously. For claim 3 we use the facts that $(\mathrm{Most}^1)^\star$ is the restriction of M to the cardinalities 2^n and that the second vectorization M^2 of M is definable in FO$(\mathrm{M}, +, \times)$ (see [36]). $\square$

The following lemma can be now used.

Lemma 3.5. *Let $S \subseteq \mathbb{N}$, p a prime, and $q > 1$ relatively prime to p. Then*

1. *Q_S is definable in* $\mathrm{FO}(+, \times)$ *iff S either finite or cofinite.*
2. *D_q is not definable in* $\mathrm{FO}(\mathrm{D}_p, +, \times)$.

Proof. The first claim follows from non-definability of the language PARITY in $\mathrm{FO}(+, \times)$ [19,41] (see Theorem 4.3 in [42]). The second claim goes back to [43]. □

By combining Theorem 3.4 and Lemma 3.5 we can show the following.

Corollary 3.6. *Let $S \subseteq \mathbb{N}$, p a prime, and $q > 1$ relatively prime to p. Then*

1. *$\mathcal{Q}_S$ is definable in* $\mathrm{MSO}(+)$ *iff S is either finite or cofinite.*
2. *$\mathcal{D}_q$ is not definable in* $\mathrm{MSO}(\mathcal{D}_p, +)$.

Another corollary of Theorem 3.4 is that the quantifier Most^1 is not definable in the logic $\mathrm{MSO}(+)$.

Corollary 3.7. *The quantifier* Most^1 *is not definable in* $\mathrm{MSO}(+)$.

Proof. For a contradiction, let us assume that Most^1 is definable in $\mathrm{MSO}(+)$. By the results of [38], the quantifier I^2 can then also be defined in $\mathrm{MSO}(+)$. Now, since I^2 is numerical, Theorem 3.4 implies that the quantifier I is definable in $\mathrm{FO}(+, \times)$. This is a contradiction since

$$\mathrm{FO}(\mathrm{I}, +, \times) \equiv \mathrm{FO}(\mathrm{M}, +, \times) > \mathrm{FO}(+, \times). \qquad \square$$

It is possible to replace $\mathrm{MSO}(+)$ by SO in Theorem 3.2. The idea is that, if $\mathcal{Q}_1$ is definable in $\mathrm{SO}(\mathcal{Q}_2)$, then in the defining formula, for some k, only relations of arity at most k are quantified. We will not pursue this generalization in full generality but only consider the special case of the quantifier Most^1.

Theorem 3.8. *The quantifier* Most^1 *is not definable in* SO.

Proof. It suffices to show that Most^1 is not definable in $\mathrm{FO}(\exists^2_k)$ for any k. For a contradiction, assume that Most^1 is definable in $\mathrm{FO}(\exists^2_k)$. We will now proceed as follows: First an analogous translation as in Theorem 3.2 is used to show that definability of Most^1 in $\mathrm{FO}(\exists^2_k)$ implies that a certain padded version of the class M is definable in $\mathrm{FO}(+, \times)$ over cardinalities 2^{n^k}. This class corresponds to a variant L of the binary language MAJ

$$\mathrm{MAJ} = \{ w \in \{0,1\}^+ \mid |w|_1 > |w|_0 \},$$

when ordered $\{P\}$-structures are viewed as binary strings. Definability of L in $\mathrm{FO}(+, \times)$ would allow us to construct constant depth quasi-polynomial size ($2^{\log(n)^{O(1)}}$) AND/OR circuits for MAJ contradicting the result of [20] and [21].

We will now discuss the proof in more detail. Note first that, in order to translate the quantifier $\exists^2_k$ to the logic $\mathrm{FO}(+, \times)$, we need to redefine the structure $\hat{\mathfrak{A}}$ (see Definition 3.1) to have a domain of the form $\{0, \ldots, 2^{n^k} - 1\}$ instead of $\{0, \ldots, 2^n - 1\}$. In other respects the definition of $\hat{\mathfrak{A}}$ is not altered. We can now use the fact that there is a one-to-one correspondence between integers $m \in \{0, \ldots, 2^{n^k} - 1\}$ and k-ary relations R of $\{0, \ldots, n-1\}$. In other words, by using the lexicographic ordering on k-tuples, a relation R can be encoded by a binary string of length n^k corresponding to the binary representation of a unique integer $m < 2^{n^k}$. It is straightforward to adjust the translation in the proof of Theorem 3.2 to this setting. The only difference is that the unary relations Y_i are now k-ary. The translation is modified as follows to translate the k-ary atomic formulas $Y_i(x_1, \ldots, x_k)$:

$$Y_i(x_1, \ldots, x_k) \quad \rightsquigarrow \quad \exists z\big(\mathrm{BIT}\big(y_i, n^k - (z+1)\big) \wedge z = n^{k-1}x_1 + \cdots + nx_{k-1} + x_k\big).$$

We assumed that the quantifier Most^1 is definable in $\mathrm{FO}(\exists^2_k)$ which now implies that the following class $(\mathrm{Most}^1)^\star$

$$\big(\mathrm{Most}^1\big)^\star = \big\{(B, P, <, +, \times) \bigm| (B, P) = \hat{\mathfrak{A}} \text{ and } \mathfrak{A} \in \mathrm{Most}^1\big\}$$

can be defined in the logic $\mathrm{FO}(+, \times)$. Note that $(B, P, <, +, \times) \in (\mathrm{Most}^1)^\star$ iff $B = \{0, \ldots, 2^{n^k} - 1\}$, $P \subseteq \{0, \ldots, 2^n - 1\}$, and $|P| > 2^{n-1}$. By viewing the structures of $(\mathrm{Most}^1)^\star$ as binary words, it follows that the binary language L

$$L = \{w_1 w_2 \in \{0,1\}^*\colon |w_1| = 2^n,\ w_1 \in \mathrm{MAJ},\ |w_2| = 2^{n^k} - 2^n,\ w_2 \in 0^*\},$$

can be defined in the logic $\mathrm{FO}(+, \times)$.

Since $\mathrm{FO}(+, \times)$ corresponds to DLOGTIME-uniform AC^0, we get that there is a uniform family $(C_n)_{n\in\mathbb{N}}$ of constant depth polynomial size AND/OR circuits accepting L. These circuits can be now used to construct a family $(C'_{2^n})_{n\in\mathbb{N}}$ of constant depth quasi-polynomial size AND/OR circuits for MAJ in input lengths 2^n: the circuit C'_{2^n} for length 2^n binary words is acquired from the circuit $C_{2^{n^k}}$ by turning the input gates with index i, for $2^n < i \leqslant 2^{n^k}$, to constant 0 gates. It is easy to see that the size of C'_m is $2^{O(\log(m)^k)}$. This is a contradiction with the results of [20] and [21] showing that such a quasi-polynomial size family $(C'_{2^n})_{n\in\mathbb{N}}$ cannot exist. □

4. Conclusion

We have shown that definability of second-order generalized quantifiers can be reduced to definability of first-order generalized quantifiers. We have indicated a couple of corollaries to our characterization but surely there is more to be done here, e.g., with replacing the base logic $\mathrm{MSO}(+)$ by SO as in Theorem 3.8. In particular, Theorem 3.8 solves the open problem proposed in [24,25], where we studied the collective meanings of natural language quantifiers. It suggests, as we argued in [24], that the type-shifting strategy [27] to define the meanings of natural language quantification might be too restricted in its computational power. It is likely that second-order logic is not enough to capture natural language semantics. Another interpretation would be that everyday language does not realize hard collective quantifiers (for sure they are marginal at best) due to their complexity.

Acknowledgments

The authors would like to thank Heribert Vollmer for valuable comments. The first author was supported by grant 264917 of the Academy of Finland. The second author was supported by NWO Veni grant 639.021.232.

References

[1] A. Mostowski, On a generalization of quantifiers, Fundam. Math. 44 (1957) 12–36.
[2] P. Lindström, First order predicate logic with generalized quantifiers, Theoria 32 (1966) 186–195.
[3] J. Väänänen, Generalized quantifiers, an introduction, in: Generalized Quantifiers and Computation, Aix-en-Provence, 1997, in: Lecture Notes in Comput. Sci., vol. 1754, Springer, Berlin, 1999, pp. 1–17.
[4] H.-D. Ebbinghaus, J. Flum, Finite Model Theory, 2nd edition, Perspectives in Mathematical Logic, Springer-Verlag, 1999.
[5] S. Peters, D. Westerståhl, Quantifiers in Language and Logic, Clarendon Press, Oxford, 2006.
[6] R. Fagin, Generalized first-order spectra and polynomial-time recognizable sets, in: Complexity of Computation, Proc. SIAM-AMS Sympos. Appl. Math., New York, 1973, in: SIAM–AMS Proc., vol. VII, Amer. Math. Soc., Providence, RI, 1974, pp. 43–73.
[7] H.-J. Burtschick, H. Vollmer, Lindström quantifiers and leaf language definability, Int. J. Found. Comput. Sci. 9 (3) (1998) 277–294.
[8] A. Andersson, On second-order generalized quantifiers and finite structures, Ann. Pure Appl. Log. 115 (1–3) (2002) 1–32.
[9] J. Kontinen, Definability of second order generalized quantifiers, PhD thesis, University of Helsinki, 2005.
[10] J. Kontinen, Definability of second order generalized quantifiers, Arch. Math. Log. 49 (3) (2010) 379–398.
[11] J. Kontinen, The hierarchy theorem for second order generalized quantifiers, J. Symb. Log. 71 (1) (2006) 188–202.
[12] D.P. Bovet, P. Crescenzi, R. Silvestri, A uniform approach to define complexity classes, Theor. Comput. Sci. 104 (2) (1992) 263–283.
[13] T. Peichl, H. Vollmer, Finite automata with generalized acceptance criteria, Discrete Math. Theor. Comput. Sci. 4 (2001) 179–192.
[14] M. Galota, H. Vollmer, A generalization of the Büchi–Elgot–Trakhtenbrot-theorem, in: Computer Science Logic, Lecture Notes in Computer Science, Springer-Verlag, Berlin, Heidelberg, 2001, pp. 355–368.
[15] J.R. Büchi, C.C. Elgot, Decision problems of weak second order arithmetics and finite automata, Part I, Not. Am. Math. Soc. 5 (1958) 834.
[16] J.R. Büchi, On a decision method in restricted second-order arithmetic, in: Proceedings Logic, Methodology and Philosophy of Sciences 1960, Stanford University Press, Stanford, CA, 1962, pp. 1–11.
[17] B.A. Trakhtenbrot, Finite automata and logic of monadic predicates, Dokl. Akad. Nauk SSSR 140 (1961) 326–329, in Russian.
[18] M. More, F. Olive, Rudimentary languages and second-order logic, Math. Log. Q. 43 (3) (1997) 419–426.
[19] M.L. Furst, J.B. Saxe, M. Sipser, Parity, circuits, and the polynomial-time hierarchy, Math. Syst. Theory 17 (1) (1984) 13–27, http://dx.doi.org/10.1007/BF01744431.
[20] A.C.-C. Yao, Separating the polynomial-time hierarchy by oracles, in: Annual IEEE Symposium on Foundations of Computer Science, 1985, pp. 1–10.
[21] J.T. Håstad, Computational Limitations of Small-Depth Circuits, MIT Press, Cambridge, MA, 1987.
[22] J. Torán, Structural properties of the counting hierarchies, PhD thesis, Facultat d'Informatica de Barcelona, Barcelona, Spain, 1988.
[23] H. Vollmer, Relating polynomial time to constant depth, Theor. Comput. Sci. 207 (1) (1998) 159–170.
[24] J. Kontinen, J. Szymanik, A remark on collective quantification, J. Log. Lang. Inf. 17 (2) (2008) 131–140.
[25] J. Szymanik, Quantifiers in time and space. Computational complexity of generalized quantifiers in natural language, PhD thesis, University of Amsterdam, 2009.
[26] J. van der Does, Sums and quantifiers, Linguist. Philos. 16 (5) (1993) 509–550.
[27] Y. Winter, Flexibility Principles in Boolean Semantics, The MIT Press, London, 2001.
[28] J. Szymanik, M. Zajenkowski, Comprehension of simple quantifiers. Empirical evaluation of a computational model, Cogn. Sci. 34 (3) (2010) 521–532.
[29] J. Szymanik, Computational complexity of polyadic lifts of generalized quantifiers in natural language, Linguist. Philos. 33 (2010) 215–250.
[30] F. Schlotterbeck, O. Bott, Easy solutions for a hard problem? The computational complexity of reciprocals with quantificational antecedents, J. Log. Lang. Inf. 22 (4) (2013) 363–390.
[31] J.T. Lønning, Plurals and collectives, in: J. van Benthem, A. ter Meulen (Eds.), Handbook of Logic and Language, 2nd edition, Elsevier, 2011, pp. 989–1035.
[32] M. Mostowski, J. Szymanik, Semantic bounds for everyday language, Semiotica 188 (1–4) (2012) 363–372.
[33] N. Immerman, Descriptive Complexity, Graduate Texts in Computer Science, Springer-Verlag, New York, 1999.

[34] L.J. Stockmeyer, The polynomial-time hierarchy, Theor. Comput. Sci. 3 (1) (1976) 1–22, 1977.
[35] I. Parberry, G. Schnitger, Parallel computation with threshold functions, J. Comput. Syst. Sci. 36 (3) (1988) 278–302.
[36] D.A.M. Barrington, N. Immerman, H. Straubing, On uniformity within NC^1, J. Comput. Syst. Sci. 41 (3) (1990) 274–306.
[37] J. Kontinen, H. Niemistö, Extensions of MSO and the monadic counting hierarchy, Inf. Comput. 209 (1) (2011) 1–19.
[38] J. Kontinen, A logical characterization of the counting hierarchy, ACM Trans. Comput. Log. 10 (1) (2009).
[39] L. Hella, J. Kontinen, K. Luosto, Regular representations of uniform TC^0, Institut Mittag-Leffler Preprint Series (The Program of Fall 2009), 2010.
[40] W. Hesse, E. Allender, D.A.M. Barrington, Uniform constant-depth threshold circuits for division and iterated multiplication, J. Comput. Syst. Sci. 65 (4) (2002) 695–716, special issue on complexity, 2001 (Chicago, IL).
[41] M. Ajtai, Σ_1^1-formulae on finite structures, Ann. Pure Appl. Log. 24 (1) (1983) 1–48, http://dx.doi.org/10.1016/0168-0072(83)90038-6.
[42] D.A.M. Barrington, N. Immerman, C. Lautemann, N. Schweikardt, D. Thérien, First-order expressibility of languages with neutral letters or: the Crane Beach conjecture, J. Comput. Syst. Sci. 70 (2) (2005) 101–127.
[43] R. Smolensky, Algebraic methods in the theory of lower bounds for boolean circuit complexity, in: STOC, 1987, pp. 77–82.