

A Dichotomy Result for Ramsey Quantifiers^{*}

Ronald de Haan¹ and Jakub Szymanik²

¹ Algorithms and Complexity Group, Vienna University of Technology
dehaan.ronald@gmail.com

² Institute for Logic, Language and Computation, University of Amsterdam
jakub.szymanik@gmail.com

Abstract. Ramsey quantifiers are a natural object of study not only for logic and computer science, but also for formal semantics of natural language. Restricting attention to finite models leads to the natural question whether all Ramsey quantifiers are either polynomial-time computable or NP-hard, and whether we can give a natural characterization of the polynomial-time computable quantifiers. In this paper, we first show that there exist intermediate Ramsey quantifiers and then we prove a dichotomy result for a large and natural class of Ramsey quantifiers, based on a reasonable and widely-believed complexity assumption. We show that the polynomial-time computable quantifiers in this class are exactly the constant-log-bounded Ramsey quantifiers.

1 Motivations

Traditionally, definability questions have been a mathematical core of natural language semantics. For example, over the years in generalized quantifier theory efforts have been directed to classify quantifier constructions with respect to their expressive power (see [1] for an extensive overview). Another already classical feature of the theory is searching for linguistic and later computer-science applications. That is one of the reasons to often investigate quantifiers over finite models. This leads naturally to questions about computational complexity. In [2, 3] it has been observed that some natural language sentences, when assuming their branching interpretation, are NP-complete. Sevenster has also proved a dichotomy theorem for independent-friendly quantifier prefixes that can capture branching quantification, namely they are either decidable in LOGSPACE or NP-hard [4]. Following this line of research, Szymanik [6, 5] searched for more natural classes of intractable generalized quantifiers. He found out that some reciprocal sentences with quantified antecedents under the strong interpretation (see [7]) define NP-complete classes of finite models.³ He has also noted that all known examples of semantic intractability have a common source: they all

^{*} The first author was supported by the European Research Council (ERC), project 239962, and the Austrian Science Fund (FWF), project P26200. The second author was supported by Veni grant NWO 639.021.232.

³ These results have interestingly also found empirical interpretations, see [8, 9].

express Ramsey-like properties [6]. Hence, he asked for a dichotomy theorem for Ramsey quantifiers.

In this technical paper we first show that there exist intermediate Ramsey quantifiers and then we display a dichotomy result for a large class of Ramsey quantifiers, based on a reasonable and widely-believed complexity assumption. Namely, we show that the Ramseyfication of polynomial-time and constant-log-bounded monadic quantifiers result in polynomial time computable Ramsey quantifiers while assuming the Exponential Time Hypothesis. The notion of constant-log-boundedness is a version of the boundedness condition known from finite-model theory literature [10], where the bound on the upper side is replaced by $c \log n$. As the property of boundedness plays an important role in definability theory of polyadic quantifiers [11], we conclude by asking whether the new notion of constant-log-boundedness gives rise to some interesting descriptive results.

2 Preliminaries

2.1 Generalized Quantifiers

Generalized quantifiers might be defined as classes of models (see e.g. [1]). The formal definition is as follows:

Definition 1 ([12]). *Let $t = (n_1, \dots, n_k)$ be a k -tuple of positive integers. A generalized quantifier of type t is a class Q of models of a vocabulary $\tau_t = \{R_1, \dots, R_k\}$, such that R_i is n_i -ary for $1 \leq i \leq k$, and Q is closed under isomorphisms, i.e. if M and M' are isomorphic, then*

$$(M \in Q \iff M' \in Q).$$

Finite models can be encoded as finite strings over some vocabulary, hence, we can easily fit the notions into the descriptive complexity paradigm (see e.g. [13]):

Definition 2. *By the complexity of a quantifier Q we mean the computational complexity of the corresponding class of finite models, that is, the complexity of deciding whether a given finite model belongs to this class.*

For some interesting early results on the computational complexity of various forms of quantification, see [14].

2.2 Computational Complexity

Problems in NP that are neither in P nor NP-complete are called NP-intermediate, and the class of such problems is called NPI. Ladner [15] proved the following seminal result:

Theorem 1. *If $P \neq NP$, then NPI is not empty.*

Assuming $P \neq NP$, Ladner constructed an artificial NPI problem. Schaefer [16] proved a dichotomy theorem for Boolean constraint satisfaction, thereby providing conditions under which classes of Boolean constraint satisfaction problems can not be in NPI. It remains an interesting open question whether there are some natural problems in NPI [17].

The Exponential Time Hypothesis The Exponential Time Hypothesis (ETH) says that 3-SAT (or any of several related NP-complete problems) cannot be solved in subexponential time in the worst case [18]. The ETH implies that $P \neq NP$. It also provides a way to obtain lower bounds on the running time of algorithms solving certain fundamental computational problems [19].

Definition 3 (Definition 3.22 and Lemma 3.23 in [20]). *Let $f, g : \omega \rightarrow \omega$ be computable functions. Then $f \in o(g)$ (also denoted $f(n) \in o(g(n))$) if there is a computable function h such that for all $\ell \geq 1$ and $n \geq h(\ell)$, we have:*

$$f(n) \leq \frac{g(n)}{\ell}.$$

Alternatively, the following definition is equivalent. We have that $f \in o(g)$ if there exists $n_0 \in \omega$ and a computable function $\iota : \omega \rightarrow \omega$ that is nondecreasing and unbounded such that for all $n \geq n_0$, it holds that $f(n) \leq \frac{g(n)}{\iota(n)}$.

Intuitively, if a function $f(n)$ is $o(g(n))$, it means that $g(n)$ grows faster than $f(n)$, when the values for n get large enough.

Exponential Time Hypothesis:

3-SAT cannot be solved in time $2^{o(n)}$, where n denotes the number of variables in the input formula.

The following result, that we use to prove the existence of intermediate Ramsey quantifiers is an example of a lower bound based on the ETH. For the problem k -CLIQUE, the input is a simple graph $G = (V, E)$ and a positive integer k . The question is whether G contains a clique of size k .

Theorem 2 ([21]). *Assuming the ETH, there is no $f(k)m^{o(k)}$ time algorithm for k -CLIQUE, where m is the input size and where f is a computable function.*

3 Ramsey Quantifiers

3.1 Ramsey Theory and Quantifiers

Informally speaking the Finite Ramsey Theorem [22] states the following:

The Finite Ramsey Theorem — general schema *When coloring a sufficiently large complete finite graph, one will find a large homogeneous subset, i.e., a complete subgraph with all edges of the same color, of arbitrary large finite cardinality.*

For suitable explications of what “large set” means we obtain various Ramsey properties. For example, “large set” may mean a “set of cardinality at least $f(n)$ ”, where f is a function from natural numbers to natural numbers on a universe with n elements (see e.g. [11]). We will adopt this interpretation and study the computational complexity of the Ramsey quantifiers determined by various functions f . Note that in our setting of finite models with one binary relation Q , that we will describe below, Ramsey quantifiers are essentially equivalent to the problem of determining whether a graph has a clique of a certain size.

3.2 Basic Proportional Ramsey Quantifiers

Let us start with a precise definition of “large relative to the universe”.

Definition 4. For any rational number q between 0 and 1 we say that the set $A \subseteq U$ is q -large relative to U if and only if

$$\frac{\text{card}(A)}{\text{card}(U)} \geq q.$$

In this sense q determines the *basic proportional Ramsey quantifier* R_q of type (2).

Definition 5. Let $\mathbb{M} = (M, S)$ be a relational model with universe M and a binary relation S . We say that $\mathbb{M} \in R_q$ iff there is a q -large (relative to M) $A \subseteq M$ such that for all $a, b \in A$, $\mathbb{M} \models S(a, b)$.

Theorem 3 ([5]). For every rational number q , such that $0 < q < 1$, the corresponding Ramsey quantifier R_q is NP-complete.

3.3 Tractable Ramsey Quantifiers

We have shown some examples of NP-complete Ramsey quantifiers. In this section we will describe a class of Ramsey quantifiers computable in polynomial time. Let us start with considering an arbitrary function $f : \omega \rightarrow \omega$.

Definition 6. We say that a set $A \subseteq U$ is f -large relatively to U iff

$$\text{card}(A) \geq f(\text{card}(U)).$$

Then we define Ramsey quantifiers of type (1, 2) corresponding to the notion of “ f -large”.

Definition 7. We define R_f as the class of relational models $\mathbb{M} = (M, S)$, with universe M and a binary relation S , such that there is an f -large set $A \subseteq M$ such that for each $a, b \in A$, $\mathbb{M} \models S(a, b)$.

Notice that the above definition is very general and covers all previously defined Ramsey quantifiers. For example, we can reformulate Theorem 3 in the following way:

Corollary 1. Let $f(n) = \lceil rn \rceil$, for some rational number r such that $0 < r < 1$. Then the quantifier R_f defines a NP-complete class of finite models.

Let us put some further restrictions on the class of functions we are interested in. First of all, as we will consider f -large subsets of the universe we can assume that for all $n \in \omega$, $f(n) \leq n + 1$. In that setting the quantifier R_f says about a set A that it has at least $f(n)$ elements, where n is the cardinality of the universe. We allow the function to be equal to $n + 1$ just for technical reasons as in this case the corresponding quantifier has to be always false.

Our crucial notion goes back to paper [10] of Väänänen:

Definition 8. We say that a function f is bounded if

$$\exists m \forall n [f(n) < m \vee n - m < f(n)].$$

Otherwise, f is unbounded.

Theorem 4 ([5]). If f is polynomial-time computable and bounded, then the Ramsey quantifier R_f is polynomial-time computable.

Proof (sketch). Let m be the integer such that for all n it holds that either $f(n) < m$ or $n - m < f(n)$. This means that for every model $\mathbb{M} = (M, Q)$ with $|M| = n$, to decide if $\mathbb{M} \in R_f$, we only need to consider those subsets $A \subseteq M$ for which holds $|A| < m$ or $|A| > n - m$. Since m is a constant, these are only polynomially many.

3.4 Intermediate Ramsey Quantifiers

We have shown that proportional Ramsey quantifiers define NP-complete classes of finite models. On the other hand, we also observed that bounded Ramsey quantifiers are polynomial-time computable.

The first question we might ask is whether for all functions f the Ramsey quantifier R_f is either polynomial-time computable or NP-complete. We observe that this cannot be the case if we make some standard complexity-theoretic assumptions.

Intermediate Ramsey Quantifiers

Theorem 5. Let $f(n) = \lceil \log n \rceil$. The quantifier R_f is neither polynomial-time computable nor NP-complete, unless the ETH fails.

Proof. Firstly assume that R_f is NP-complete. This means that there is a polynomial-time reduction R from 3-SAT to R_f (that takes as input an instance of 3-SAT with n variables and produces an equivalent instance of R_f with n^d elements, for some constant d). There is a straightforward brute force search algorithm A that solves R_f in time $O(n^{f(n)}) = O(n^{\lceil \log n \rceil})$. Composing R and A then leads to an algorithm that solves 3-SAT in time $O((n^d)^{\lceil \log n^d \rceil}) = O(n^{d^2 \log n}) = O(2^{d^2 (\log n)^2})$, for some constant d , which runs in subexponential time. Therefore, the ETH fails.

On the other hand, it is known that if the problem of deciding whether a given graph with n vertices has a clique of size $\geq \log n$ (equivalently R_f , for $f(n) = \lceil \log n \rceil$) is solvable in polynomial time, then the ETH fails [23, Theorem 3.4].

In other words, assuming ETH, there exists Ramsey quantifiers whose model checking problem is an example of an NP-intermediate problem in computational complexity, i.e., it is a problem that is in NP but is neither polynomial-time computable nor NP-complete [15].

The remaining open question is whether there exists a natural class of functions such that under some reasonable complexity assumptions (e.g., ETH) the polynomial-time Ramsey quantifiers are exactly the bounded Ramsey quantifiers. In other words:

Problem 1. Is it the case that for every function f from some ‘natural’ class we have a dichotomy theorem, i.e., R_f is either polynomial-time computable or NP-complete?

3.5 Intractable Ramsey Quantifiers

In this section, we show for a large natural class of natural functions f that R_f is not polynomial-time computable, unless the ETH fails.

Restrictions on the Class of Functions One way in which we assume the functions f to be natural is that the value $f(n)$ is computable in time polynomial in n . From now on, we will assume that this property holds for all functions f that we consider. This assumption corresponds to restricting the attention to polynomial-time computable monadic generalized quantifiers which seems reasonable from a natural language perspective [5].

In fact, for any function f that is not polynomial-time computable, the problem R_f clearly cannot be computable in polynomial time either.

Observation 6 *Let $f : \omega \rightarrow \omega$ be a function that is not polynomial-time computable. Then R_f is not polynomial-time computable.*

Proof. Let f be non polynomial-time computable function and assume for contradiction that R_f is polynomial-time computable. Take $\mathbb{M} = (U, E)$ with $|U| = n$. Find the smallest complete $A \subseteq U$ that will make $\mathbb{M} \in R_f$. Clearly that can be done in polynomial time but then $f(n) = |A|$ would be also computable in polynomial time.

Considering this observation, in the remainder of the paper we will only consider functions f that are polynomial-time computable.

Assumption 7 *The functions f that we consider are polynomial-time computable, i.e., for every $n \in \omega$, the value $f(n)$ is computable in time polynomial in n .*

Intractability Based on the ETH In this section, we set out to prove the technical results that will give us the dichotomy result for R_f , for the class of polynomial-time computable functions f . We start with considering the following class of sublinear functions.

Definition 9 (Sublinear functions). *Let $f : \omega \rightarrow \omega$ be a nondecreasing function. We say that f is sublinear if $f(n)$ is $o(n)$, i.e., if there exists some computable function $s(n)$ that is nondecreasing and unbounded, and some $n_0 \in \omega$, such that for all $n \in \omega$ with $n \geq n_0$ it holds that $f(n) \leq \frac{n}{s(n)}$.*

In order to illustrate this concept, we give a few examples of sublinear functions.

Example 1. Consider the function $f_1(n) = \lceil \log n \rceil$. This function is sublinear, which is witnessed by $s(n) = n/\lceil \log n \rceil$. Additionally, any function $f(n)$ that satisfies that $f(n) \leq \lceil \log n \rceil$, for all $n \in \omega$, is also sublinear. Next, the function $f_2(n) = \lceil \sqrt{n} \rceil$ is also sublinear, which is witnessed by $s(n) = \sqrt{n}/2$. As a final example, consider the function $f_3(n) = \lceil n/\log n \rceil$. Clearly, by taking $s(n) = \log n/2$, we get that $f_3(n) \leq n/s(n)$. Therefore, f_3 is also sublinear.

Lemma 1. *Let $f : \omega \rightarrow \omega$ be a nondecreasing function that is $o(n)$, and let $b \in \omega$ be a positive integer. Moreover, let $G = (V, E)$ be an instance of R_f . In polynomial time, we can produce some $b' \geq b$ and we can transform G into an equivalent instance $G' = (V', E')$ of R_f with n' vertices such that $f(n') \leq b'$.*

Proof (sketch). If $f(n) \leq b$, we can let $G' = G$. Therefore, assume that $f(n) > b$. We will increase n and b , by adding a polynomial number of ‘dummy’ vertices that are connected to all other vertices (and increasing b by an equal amount). It is straightforward to see that such a transformation results in an equivalent instance. Since s is nondecreasing and unbounded, we know there exists some $n_0 \in \omega$ such that for all $n \geq n_0$ it holds that $s(n) \geq 2$. Now, we define the function $\delta(n) = n + n_0$. Clearly, δ is polynomial-time computable. We show that for all $n, b \in \omega$ it holds that $f(n + \delta(n)) \leq b + \delta(n)$:

$$\begin{aligned} f(n + \delta(n)) &= f(2n + n_0) \leq \frac{2n + n_0}{s(2n + n_0)} \leq \frac{2n + n_0}{2} = n + \frac{n_0}{2} \\ &\leq n + n_0 \leq b + n + n_0 = b + \delta(n). \end{aligned}$$

Now, let $b' = b + \delta(n)$. Then, if we add $\delta(n)$ many vertices to G that are connected to all other vertices, we get an instance G' with n' vertices such that $f(n') \leq b'$.

Proposition 1. *Let $f : \omega \rightarrow \omega$ be a nondecreasing unbounded function that is $o(n)$. Then R_f is not solvable in polynomial time, unless the ETH fails.*

Proof. In order to prove our result, we will assume that R_f is solvable in polynomial time, and then show that the ETH fails. In particular, we will show that k -CLIQUE is solvable in time $f'(k)m^{o(k)}$, which implies the failure of the ETH by Theorem 2.

Firstly, we will define a function f^{-1} as follows. We let:

$$f^{-1}(h) = \min\{q : f(q) \geq h\}.$$

Since f is an unbounded nondecreasing function, we get that f^{-1} is an unbounded nondecreasing function as well.

We give an algorithm that solves k -CLIQUE in the required amount of time. Let (G, k) be an instance of k -CLIQUE, where $G = (V, E)$ is a graph with n vertices. Let m denote the size of G (in bits). Intuitively, we will add exactly the right number of ‘dummy’ vertices to G , resulting in a graph $G' = (V', E')$,

to make sure that $f(n') = k$ where $n' = |V'|$ (while ensuring that G has a k -clique if and only if G' has a k -clique). To be more precise, we will construct a number k' such that $f(n') = k'$ and such that G has a k -clique if and only if G' has a k' -clique. Consider the number $q = f^{-1}(k)$, and define $\ell = f(q) - k$. By definition of f^{-1} , we know that $f(f^{-1}(k)) \geq k$, and thus that $\ell \geq 0$. We may assume without loss of generality that $\ell \leq q - n$ and thus that $0 \leq \ell \leq q - n$. If this were not the case, we could invoke Lemma 1 (by taking $b = q - n$) to increase q to a number q' (and update ℓ to ℓ' accordingly) such that $q' - n \geq q - n \geq f(q) \geq f(q') - k = \ell'$.

We now construct G' from G by adding $q - n$ many new vertices, where ℓ of them are connected in G' to all existing vertices in G , and the remaining new vertices are not connected to any other vertex. We then get that $n' = q$, and we let $k' = f(n') = f(q)$. It is now straightforward to verify that G has a k -clique if and only if G' has a k' -clique, and that the size of G' is at most $f^{-1}(k)m^c$ for some constant c .

Now that we constructed G' , we can use our polynomial-time algorithm to check whether $G' \in R_f$, which is the case if and only if $(G, k) \in k\text{-CLIQUE}$. This takes an amount of time that is polynomial in the size m' of G' . Since $m' \leq f^{-1}(k)m^c$ for some constant c , the combined algorithm of producing G' and deciding whether $G' \in R_f$ takes time $f'(k)(m)^{c'}$ for some function f' and some constant c' . From this we can conclude that $k\text{-CLIQUE}$ is solvable in time $f'(k)m^{c'} = f'(k)m^{o(k)}$. Therefore, by Theorem 2, the ETH fails.

We point out that the result of Proposition 1 actually already follows from a known result [24, Theorem 5.7]. For the sake of clarity, we included a self-contained proof of this statement anyway.

The class of sublinear functions as considered in the result of Proposition 1, also contains those functions f such that $f(n) \leq n^\epsilon$, for some constant ϵ such that $0 < \epsilon < 1$.

Corollary 2. *Let $f : \omega \rightarrow \omega$ be a unbounded, computable function such that for all $n \in \omega$, $f(n) \leq n^\epsilon$ for some constant rational number ϵ such that $0 < \epsilon < 1$. Then R_f is not polynomial-time computable, unless the ETH fails.*

Proof. Since $f(n) \leq n^\epsilon$, we know that $f(n) \leq n/n^{1-\epsilon}$. Then, because $s(n) = n^{1-\epsilon}$ is a nondecreasing, unbounded computable function, we can apply Proposition 1 to obtain the intractability of R_f .

Next, we turn to another class of polynomial-time computable functions f for which R_f is not polynomial-time computable unless the ETH fails.

Proposition 2. *Let $f : \omega \rightarrow \omega$ be a polynomial-time computable function such that, for sufficiently large n , it holds that $f(n) \leq n - \log n \cdot s(n)$, for some nondecreasing and unbounded computable function s . Then R_f is not polynomial-time solvable, unless the ETH fails.*

Proof. We show that a polynomial time algorithm to decide R_f can be used to show that deciding whether a given simple graph (with n vertices) contains a clique of a given size m can be solved in subexponential time, i.e., in time $2^{o(n)} \text{poly}(|G|)$. This, in turn, implies the failure of the ETH [18].

Let $G = (V, E)$ be a simple graph with n vertices. Moreover, let m be a positive integer. We will add a certain number, ℓ , of vertices to this graph, to obtain a new graph G' . We will do this in such a way that almost all of these new vertices (ℓ' of them) are connected to all other vertices. Moreover, we will make sure that $m + \ell \geq f(n + \ell)$. Then we can choose ℓ' in such a way that $m + \ell' = f(n + \ell)$. This allows us to use the polynomial time algorithm for R_f to decide whether G contains a clique of size m , since any clique of size $m + \ell'$ in G' corresponds to a clique of size m in G .

We define the nondecreasing, unbounded function t (representing the ‘inverse’ of $s(n) \log n$) as follows. Let $t(n) = \max\{h : s(h) \log h \leq n\}$. Since $s(n) \log n$ grows strictly faster than $\log n$, we get that $t(n)$ is subexponential, i.e., $t(n)$ is $2^{o(n)}$. Then, in order to ensure that $m + \ell \geq n + \ell - s(n + \ell) \log(n + \ell)$, we need that $s(n + \ell) \log(n + \ell) \geq n - m$, and thus that $n + \ell \geq t(n - m)$. This allows us to choose $\ell = t(n - m) - n = 2^{o(n)} - n$. Therefore, our reduction to R_f runs in subexponential-time. Consequently, if we were to compose this reduction and the (hypothetical) polynomial time algorithm R_f , we could decide whether G has a clique of size m in subexponential time, and thus the ETH fails.

On the other hand, there are functions f that are not bounded, but for which R_f is polynomial-time computable. Consider the function $f(n) = n - c \lceil \log n \rceil$, where c is some fixed constant. Clearly, this function f is not bounded (in the sense of Definition 8). We show that for functions of this kind, the Ramsey quantifier R_f is polynomial-time computable.

Proposition 3. *Let $c \in \omega$ be a constant, and let $f : \omega \rightarrow \omega$ be any polynomial-time computable function such that, for sufficiently large n , $f(n) \geq n - c \lceil \log n \rceil$. Then R_f is polynomial-time computable.*

Proof. Firstly, we consider the problem of, given a simple graph $G = (V, E)$ with n vertices, and an integer k , deciding whether G contains a clique of size at least $n - k$. We know that this problem can be solved in time $2^k \cdot \text{poly}(n)$ [20, Proposition 4.4]. In other words, deciding whether a graph with n vertices contains a clique of size ℓ can be done in time $2^{n-\ell} \cdot \text{poly}(n)$. We will use this result to show polynomial-time computability of R_f .

Let \mathbb{M} be a structure with a universe M containing n elements, and let $R_f xy \varphi(x, y)$ be an R_f -quantified formula. We construct the graph $G = (V, E)$ as follows. We let $V = M$, and for each $a, b \in M$ we let E contain an edge between a and b if and only if $\mathbb{M} \models \varphi(a, b)$. Clearly, G can be constructed in polynomial time.

Moreover, G has a clique of size $f(n)$ if and only if $\mathbb{M} \models R_f xy \varphi(x, y)$. Therefore, it suffices to decide whether G has a clique of size $f(n)$. We know that $f(n) \geq n - c \lceil \log n \rceil$. As mentioned above, we know we can decide this

in time $2^{n-f(n)} \cdot \text{poly}(n)$. Because $n - f(n) \leq c \lceil \log n \rceil$, we get that $2^{n-f(n)} \leq 2^{c \lceil \log n \rceil} \leq (2n)^c$. Thus, we can solve the problem in polynomial time.

Combining Theorem 4, Observation 6 and Propositions 1, 2 and 3, we get the following dichotomy result. In order to state this result, we define a notion of boundedness that differs from the one in Definition 8.

Definition 10 (Constant-log-boundedness). *Let $f : \omega \rightarrow \omega$ be a computable function. We say that f is constant-log-bounded if one of the following holds:*

- for all $n \in \omega$, $f(n)$ is bounded by a constant, i.e., there is some $m \in \omega$ such that for all $n \in \omega$ it holds that $f(n) \leq m$; or
- for all $n \in \omega$, $f(n)$ differs from n by at most $c \log n$, where c is some constant, i.e., there is some $c \in \omega$ such that for all $n \in \omega$ it holds that $f(n) \geq n - c \log n$.

Theorem 8. *Let $f : \omega \rightarrow \omega$ be a computable function. Then, assuming the ETH, R_f is polynomial-time computable if and only if f is polynomial-time computable and constant-log-bounded.*

4 Conclusions and Outlook

We investigated the computational complexity of Ramsey quantifiers. We pointed out some natural tractable (i.e., bounded) and intractable (e.g., proportional) Ramsey quantifiers. These results motivate the search for a dichotomy theorem for Ramsey quantifiers. As a next step, assuming the ETH, we showed that there exist intermediate Ramsey quantifiers (that is, Ramsey quantifiers that are neither polynomial-time computable nor NP-hard). This led to the question whether there exists a natural class of functions, and a notion of boundedness, for which (under reasonable complexity assumptions) the polynomial-time Ramsey quantifiers are exactly the bounded Ramsey quantifiers. We showed that this is indeed the case. Our main result states that assuming the ETH, a Ramsey quantifier is polynomial-time computable if and only if it corresponds to a polynomial-time computable and constant-log-bounded function.

Let us conclude with the following more logical question. The classical property of boundedness plays a crucial role in the definability of polyadic generalized quantifiers. Hella, Väänänen, and Westerståhl in [11] have shown that the Ramseyfication of Q is definable in $\text{FO}(Q)$ if and only if Q is bounded. Moreover, in a similar way, defining “joint boundedness” for pairs of quantifiers Q_f and Q_g one can notice that $\text{Br}(Q_f, Q_g)$ is definable in $\text{FO}(Q_f, Q_g)$ [11] and, therefore, polynomial-time computable for polynomial functions f and g . In this paper we substitute the boundedness definition with the notion of constant-log-boundedness, where the bound on the upper side is replaced by $c \log n$. A natural direction for future research is whether this change leads to interesting descriptive results.

References

1. Peters, S., Westerståhl, D.: Quantifiers in Language and Logic. Clarendon Press, Oxford (2006)
2. Mostowski, M., Wojtyniak, D.: Computational complexity of the semantics of some natural language constructions. *Annals of Pure and Applied Logic* **127**(1-3) (2004) 219–227
3. Sevenster, M.: Branches of Imperfect Information: Logic, Games, and Computation. PhD thesis, University of Amsterdam (2006)
4. Sevenster, M.: Dichotomy result for independence-friendly prefixes of generalized quantifiers. *The Journal of Symbolic Logic* **79**(4) (2014) 1224–1246
5. Szymanik, J.: Computational complexity of polyadic lifts of generalized quantifiers in natural language. *Linguistics and Philosophy* **33** (2010) 215–250
6. Szymanik, J.: Quantifiers in TIME and SPACE. Computational Complexity of Generalized Quantifiers in Natural Language. PhD thesis, University of Amsterdam, Amsterdam (2009)
7. Dalrymple, M., Kanazawa, M., Kim, Y., Mchombo, S., Peters, S.: Reciprocal expressions and the concept of reciprocity. *Linguistics and Philosophy* **21** (1998) 159–210
8. Schlotterbeck, F., Bott, O.: Easy solutions for a hard problem? The computational complexity of reciprocals with quantificational antecedents. *Journal of Logic, Language and Information* **22**(4) (2013) 363–390
9. Thorne, C., Szymanik, J.: Semantic complexity of quantifiers and their distribution in corpora. In: Proceedings of the International Conference on Computational Semantics. (2015)
10. Väänänen, J.: Unary quantifiers on finite models. *Journal of Logic, Language and Information* **6**(3) (1997) 275–304
11. Hella, L., Väänänen, J., Westerståhl, D.: Definability of polyadic lifts of generalized quantifiers. *Journal of Logic, Language and Information* **6**(3) (1997) 305–335
12. Lindström, P.: First order predicate logic with generalized quantifiers. *Theoria* **32** (1966) 186–195
13. Immerman, N.: Descriptive Complexity. Texts in Computer Science. Springer, New York, NY (1998)
14. Blass, A., Gurevich, Y.: Henkin quantifiers and complete problems. *Annals of Pure and Applied Logic* **32** (1986) 1–16
15. Ladner, R.E.: On the structure of polynomial time reducibility. *J. ACM* **22**(1) (1975) 155–171
16. Schaefer, T.J.: The complexity of satisfiability problems. In: Proceedings of the Tenth Annual ACM Symposium on Theory of Computing. STOC '78, New York, NY, USA, ACM (1978) 216–226
17. Grädel, E., Kolaitis, P.G., Libkin, L., Marx, M., Spencer, J., Vardi, M.Y., Venema, Y., Weinstein, S.: Finite Model Theory and Its Applications. Texts in Theoretical Computer Science. An EATCS Series. Springer (2007)
18. Impagliazzo, R., Paturi, R.: On the complexity of k-SAT. *Journal of Computer and System Sciences* **62**(2) (2001) 367–375
19. Lokshtanov, D., Marx, D., Saurabh, S.: Lower bounds based on the exponential time hypothesis. *Bulletin of the EATCS* **105** (2011) 41–72
20. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer, Berlin (2006)
21. Chen, J., Chor, B., Fellows, M., Huang, X., Juedes, D., Kanj, I.A., Xia, G.: Tight lower bounds for certain parameterized np-hard problems. *Information and Computation* **201**(2) (2005) 216 – 231

22. Ramsey, F.: On a problem of formal logic. In: Proceedings of the London Mathematical Society. Volume 30 of 2. (1929) 338–384
23. Cai, L., Juedes, D., Kanj, I.: The inapproximability of non-NP-hard optimization problems. *Theoretical Computer Science* **289**(1) (2002) 553 – 571
24. Chen, J., Huang, X., Kanj, I.A., Xia, G.: Strong computational lower bounds via parameterized complexity. *Journal of Computer and System Sciences* **72**(8) (2006) 1346–1367