Cognitive Science 34 (2010) 521–532

ISSN: 0364-0213 print / 1551-6709 online
DOI: 10.1111/j.1551-6709.2009.01078.x

Comprehension of Simple Quantifiers: Empirical Evaluation of a Computational Model

Jakub Szymanik,[a] Marcin Zajenkowski[b]

[a]*Institute for Logic, Language and Computation, University of Amsterdam*
[b]*Department of Psychology, University of Warsaw*

Received 9 December 2008; received in revised form 4 September 2009; accepted 21 September 2009

Abstract

We examine the verification of simple quantifiers in natural language from a computational model perspective. We refer to previous neuropsychological investigations of the same problem and suggest extending their experimental setting. Moreover, we give some direct empirical evidence linking computational complexity predictions with cognitive reality. In the empirical study we compare time needed for understanding different types of quantifiers. We show that the computational distinction between quantifiers recognized by finite-automata and push-down automata is psychologically relevant. Our research improves upon, the hypotheses and explanatory power of recent neuroimaging studies as well as provides evidence for the claim that human linguistic abilities are constrained by computational complexity.

Keywords: Language comprehension; Working memory; Generalized quantifiers; Finite- and push-down automata; Computational semantics of natural language

1. Introduction

We investigate the comprehension of simple quantifiers in natural language as described in a computational model posited by many linguists and logicians (see e.g., van Benthem, 1986). We refer to a recent neuropsychological investigation of the same problem by McMillan, Clark, Moore, Devita, and Grossman (2005) and account for some troubles with the interpretation of its results (see Szymanik, 2007). Moreover, we give some direct empirical evidence linking the computational complexity predictions with cognitive reality. Therefore, we provide an argument in the recent debate on the role of computational complexity in the cognitive science (see e.g., van Rooij, 2008). In particular, we compare time needed

Correspondence should be sent Jakub Szymanik, Institute for Logic, Language and Computation, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands. E-mail: j.szymanik@uva.nl

for verifying different types of quantifier sentences. We show that the computational distinction between quantifiers recognized by finite-automata (simple devices without internal memory) and push-down automata (finite-automata with data storage) is psychologically relevant.

1.1. Computability and cognition

One of the primary objectives of cognitive psychology is to explain human cognitive performance. Taking a very abstract perspective we can say that a cognitive task is a computational task. Namely, the aim of a cognitive task is to transform the initial given state of the world into some desired final state. Therefore, cognitive tasks can be identified with functions from possible initial states of the world into possible final states of the world. Notice that this understanding of cognitive tasks is very closely related to psychological practice. For instance, experimental psychology is naturally task oriented, because subjects are typically studied in the context of specific experimental tasks (see e.g., van Rooij, 2008).

Marr (1983) proposed a commonly accepted general framework for analyzing levels of explanation in cognitive sciences. In order to focus on the understanding of specific problems, he identified three levels (ordered according to decreasing abstraction):

1. computational level (problems that cognitive ability has to overcome);
2. algorithmic level (the algorithms that may be used to achieve a solution); and
3. implementation level (how it is actually done in neural activity).

Cognitive science has put a lot of effort into investigating the computational level of linguistic competence and today computational restrictions are taken very seriously when discussing cognitive capacities. For instance, a psychological version of the Church–Turing thesis (see Church, 1936; Turing, 1936)—stating that the human mind can only deal with computable problems—is widely accepted. Moreover, complexity restrictions on cognitive tasks have already been noted in the philosophy of language and mind (see e.g., Cherniak, 1981; Hofstadter, 2007), the theory of language (see e.g., Levesque, 1988; Szymanik, 2009), and psychology of vision (see e.g., Tsotsos, 1990), leading to many variants of the Tractable Cognition Thesis stating that human cognitive (linguistic) capacities are constrained by computational resources, like time and memory (see e.g., Frixione, 2001; Mostowski & Wojtyniak, 2004; van Rooij, 2008). Unfortunately, there are not enough empirical studies directly linking complexity predictions of computational models with psychological reality. One reason might be that the current debate is shaped around the question: Which computable tasks are feasible for human (bounded) agents? As a result the discussion involves references to very abstract problems of high computational complexity (NP-hard and beyond). These problems are very difficult to empirically confront with cognitive reality. Our idea in this paper is to track the links between cognitive tasks and computational complexity using simpler, less theoretically involved problems. As a result, the present research increases our empirical evidence in favor of this connection.

1.2. Computability and comprehension

In this paper we are concerned with a very basic linguistic ability of verifying quantifier sentences. In particular, we deal with the capacity of recognizing the truth-value of sentences with simple quantifiers (like ''some,'' ''an even number of,'' ''more than 7,'' ''less than half'') in finite situations illustrated by pictures. We show that a simple computational model describing the processing of such sentences is psychologically plausible with respect to reaction time predictions.

Our interest in computational models of language comprehension is natural from a theoretical point of view. There is a tradition in the philosophy of language, going back to Frege (1892), of thinking about the meaning of a sentence as *the mode of presenting* its truth-value. In modern terms we can try to explicate this idea by saying that the meaning of a sentence is an algorithm for finding its truth-value. This approach has been adopted by many theoreticians, to different degrees of explicitness, very often with a psychological motivations (see, e.g., Lambalgen & Hamm, 2005; Suppes, 1982).

1.3. Previous investigations in the area

Quantifiers have been widely treated from the perspective of cognitive psychology (see e.g., Sanford, Moxey, & Paterson, 1994). But only recently cognitive science research devoted to the computational modeling of quantifier comprehension has been published for the first time. Research presented by McMillan et al. (2005) was the first attempt to investigate the neural basis of natural language quantifiers (see also McMillan, Clark, Moore, & Grossman, 2006 for evidence on quantifier comprehension in patients with focal neurodegenerative disease, Troiani, Peelle, Clark, & Grossman, 2009 for comparison between logical [Aristotelian] and numerical [cardinal] quantifiers, and Clark & Grossman, 2007 for more general discussion). It was devoted to study brain activity during comprehension of sentences with quantifiers. Using neuroimaging methods the authors examined the pattern of neuroanatomical recruitment while subjects were judging the truth-value of statements containing natural language quantifiers. According to the authors, their results verify a particular computational model of natural language quantifier comprehension posited by several linguists and logicians. One of the authors of the present paper has challenged this statement by invoking the computational difference between elementary quantifiers and parity quantifiers (see Szymanik, 2007). The starting point of this research is this very criticism. Let us have a closer look at it.

McMillan et al. (2005) were considering the following two standard types of quantifiers: first-order and higher-order quantifiers. First-order quantifiers are those expressible in first-order predicate calculus, which is the logic containing only quantifiers $\exists$ and $\forall$ binding individual variables. In the research, the following first-order quantifiers were used: ''all,'' ''some,'' and ''at least 3.'' Higher-order quantifiers are those not expressible in first-order logic, for example, ''most,'' ''every other.'' The subjects taking part in the experiment were presented with the following higher-order quantifiers: ''less than half of,'' ''an even number of,'' ''an odd number of.''

To recognize first-order quantifiers we only need computability models, that do not use any form of internal memory (data storage). Intuitively, to check whether sentence (1) is true we do not have to involve short-term memory (working memory capacity) (for a psychological model, see, e.g., Baddeley, 2007).

(1) Every sentence in this paper is grammatically correct.

It suffices to read the sentences from this article one by one. If we find an incorrect one, then we know that the statement is false. Otherwise, if we read the entire paper without finding any incorrect sentence, then statement (1) is true. We can proceed in a similar way for other first-order quantifiers. Formally, it was proven by van Benthem (1986) that first-order quantifiers can be computed by such simple devices as finite automata without cycles (loops of length > 1).

Theorem 1. A monadic quantifier Q is first-order definable if and only if it can be recognized by an finite automaton without cycles.

For example, have a look at the automata in Figs. 1 and 2.

However, for recognizing some higher-order quantifiers, like ''less than half'' or ''most,'' we need computability models making use of internal memory. Intuitively, to check whether sentence (1) is true we must identify the number of correct sentences and hold it in working memory to compare with the number of incorrect sentences.

(2) Most of the sentences in this paper are grammatically correct.

Mathematically speaking, such an algorithm can be realized by a push-down automaton (PDA; see, e.g., Hopcroft, Motwani, & Ullman, 2000).

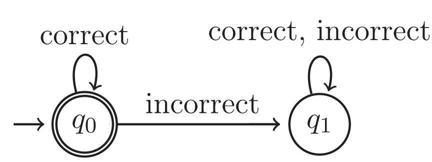

Fig. 1. This finite automaton checks whether every sentence in the text is grammatically correct. It inspects the text sentence by sentence starting in the accepting state (double circled), q_0. As long as it does not find an incorrect sentence it stays in the accepting state. If it finds such a sentence, then it already ''knows'' that the sentence is false and move to the rejecting state, q_1, where it stays no matter what sentences come next.

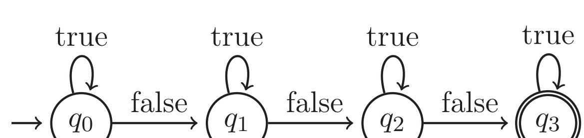

Fig. 2. This finite automaton recognizes whether at least three sentences in the text are false. This automata needs four states. It starts in the rejecting state, q_0, and eventually, if the condition is satisfied, moves to the accepting state, q_3. Furthermore, notice that to recognize ''at least eight'' we would need nine states and so on.

From the perspective of those computational differences, McMillan et al. (2005) have hypothesized that all quantifiers recruit the right inferior parietal cortex, which is associated with numerosity. Taking the distinction between the complexity of first-order and higher-order quantifiers for granted, they also predicted that only higher-order quantifiers recruit the prefrontal cortex, which is associated with executive resources, like working memory. In other words, they believe that the logical differences between first-order and higher-order quantifiers are also reflected in brain activity during processing quantifier sentences. This prediction was confirmed.

In our view the authors' interpretation of their results is not convincing. Their experimental design may not provide the best means of differentiating between the neural bases of the various kinds of quantifiers. The main point of criticism is that the distinction between first-order and higher-order quantifiers does not coincide with the computational resources required to compute the meaning of quantifiers. There is a proper subclass of higher-order quantifiers, namely divisibility (parity) quantifiers, which corresponds—with respect to memory resources—to the same computational model as first-order quantifiers. In fact, most of the quantifiers identified in the research as higher-order quantifiers can be recognized by finite automata. Both ''an even number'' and ''an odd number'' are quantifiers recognized by two-state finite automata with a transition from the first state to the second and vice versa. In general, exactly the quantifiers definable in divisibility logic, $FO(D_n)$ (i.e., first-order logic enriched by all quantifiers ''divisible by n,'' for $n \geq 2$), are recognized by finite automata (FA) (see Mostowski, 1998).

Theorem 2. A monadic quantifier Q is definable in the divisibility logic iff it can be recognized by a finite automaton.

Let us consider a relevant example. In the case of the automaton corresponding to ''even'' the initial state is also the accepting state. In the automaton for ''odd'' the other state is the accepting one. Intuitively, to check whether sentence (1) is true you do not need to count the number of false sentences and then compare it with that of the set of even integers.

(3) An even number of the sentences in this paper is false.

You need only remember parity. For example when you find a false sentence you write ''1'' at the blackboard, if you find another one you erase ''1'' and put ''0'' again, then if you see another false sentence you put ''1'' in place of ''0,'' and so on. At every moment you have only one digit at the blackboard no matter how long is the paper. Compare with the automaton from Fig. 3.

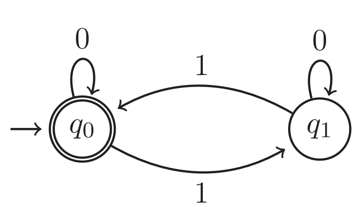

Fig. 3. Finite automaton checking whether the number of ''1''s is even.

Table 1
Quantifiers, logics, and complexity of automata

Expressibility	Examples	Recognized by
FO	''all cars,'' ''some students,'' ''at least three balls''	Acyclic FA
Parity	''an even number of balls''	FA
Proportional	''most lawyers,'' ''less than half of the students''	PDA

To sum up, first-order and higher-order quantifiers do not always differ with respect to the memory requirements. For example, ''an even number of'' is a higher-order quantifier that can still be recognized by a finite automaton. Therefore, differences in processing cannot be explained based solely on logical properties, as those are not fine grained enough. A more careful computational perspective—taking into account all mentioned results summed up in Table 1—have to be applied to investigate quantifier comprehension. In what follows we present research exploring the subject empirically with respect to the computational model outlined in this section.

1.4. The present experiment

The study compares reaction times needed for the comprehension of different types of quantifiers. In particular, it improves upon the hypothesis of McMillan et al. (2005) by taking directly into account predictions of the computational model of quantifier comprehension and not only expressibility differences among quantifiers. Additionally, we compare two classes of quantifiers inside the first-order group: Aristotelian and cardinal quantifiers, relating to the very recent research of Troiani et al. (2009).

1.4.1. General idea

First, we compared reaction time with respect to the following classes of quantifiers: those recognized by acyclic FA (first-order), those recognized by FA (parity), and those recognized by PDA (proportional). McMillan et al. (2005) did not report any data on differences between first-order and parity quantifiers.

We predict that reaction time will increase along with the computational power needed to recognize quantifiers. Hence, parity quantifiers (even, odd) will take more time than first order-quantifiers (all, some) but not as long as proportional quantifiers (less than half, more than half).

Moreover, we have additionally compared Aristotelian quantifiers with cardinal quantifiers of higher rank, for instance, ''less than eight.'' In the study of McMillan et al. (2005) only one cardinal quantifier of relatively small rank was taken into consideration, namely ''at least three.'' We predict that complexity of the mental processing of cardinal quantifiers depends on the number of states in the relevant automata. Therefore, cardinal quantifiers of high rank should be more difficult than Aristotelian quantifiers (see Fig. 2 for more explanation). Additionally, presumably the number of states in automata (size of memory needed) influences comprehension more directly than the use of loops. Hence, we hypothesize that the reaction time for the comprehension of cardinal quantifiers of higher rank is between that for parity and proportional quantifiers.

2. Method

2.1. Participants

Forty native Polish-speaking adults took part in this study. They were volunteers from the University of Warsaw undergraduate population. Nineteen of them were male and 21 were female. The mean age was 21.42 years ($SD = 3.22$) with a range of 18–30 years. Each participant was tested individually and was given a small financial reward for participation in the study.

2.2. Materials and procedure

The task consisted of 80 grammatically simple propositions in Polish containing a quantifier that probed a color feature of cars on display. For example:

Niektóre	*samochody*	*są*	*czerwone.*
Some	cars	are	red.

Mniej	*niż*	*połowa*	*samochodów*	*jest*	*niebieska.*
Less	than	half of	the cars	are	blue.

Eighty color pictures presenting a car park with cars were constructed to accompany the propositions. The colors of the cars were red, blue, green, yellow, purple, and black. Each picture contained 15 objects in two colors (see Fig. 4).

Eight different quantifiers divided into four groups were used in the study. The first group of quantifiers was first-order Aristotelian quantifiers (all, some); the second was parity quantifiers (odd, even); the third was first-order cardinal quantifiers of relatively high rank (less than eight, more than seven); and the fourth was proportional quantifiers (less than half, more than half) (see Table 2). Each quantifier was presented in 10 trials. Hence, there were in total 80 tasks in the study. The sentence matched the picture in half of the trials. Propositions with ''less than eight,'' ''more than seven,'' ''less than half,'' ''more than half'' were accompanied with a quantity of target items near the criterion for validating or falsifying the proposition. Therefore, these tasks required a precise judgment (e.g., seven targets in ''less than half''). Debriefing following the experiment revealed that none of the participants had been aware that each picture consisted of exactly 15 objects.

The experiment was divided into two parts: a short practice session followed immediately by the experimental session. Each quantifier problem resulted in a 15.5 s event. In the event the proposition and a stimulus array containing 15 randomly distributed cars were presented for 15,000 ms followed by a blank screen for 500 ms. Subjects were asked to decide if the proposition was true at the presented picture. They responded by pressing the button with

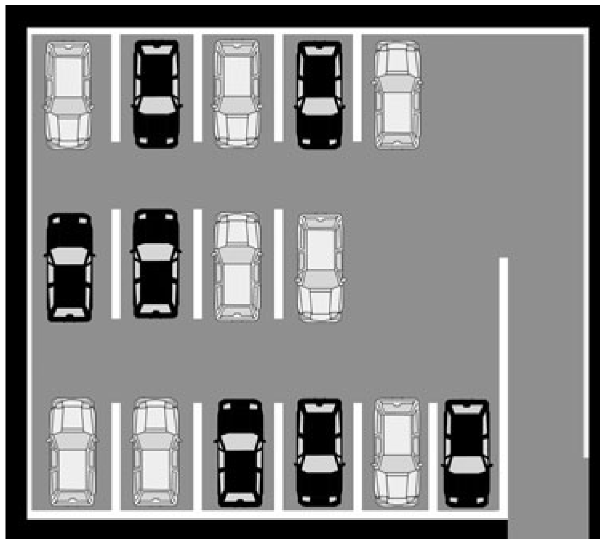

Fig. 4. An example of a stimulus used in the study.

letter ''P'' if true and the button with letter ''F'' if false. The letters refer to first letters of Polish words for ''true'' and ''false.''

The experiment was performed on a PC computer running E-Prime version 1.1.

3. Results

3.1. Analysis of accuracy

As we expected, the tasks were quite simple for our subjects and they made only a few mistakes. The percentage of correct answers for each group of quantifiers is presented in Table 2.

3.2. Comparison of reaction times

To examine the differences in means, we used repeated measures analysis of variance with type of quantifier (four levels) as the within-subject factor. The assumption of normality was verified by the Shapiro–Wilk test. Because the Mauchly's test showed violation of

Table 2
The percentage of correct answers for each group of quantifiers

Quantifier Group	Examples	Percent
Aristotelian FO	All, some	99
Parity	Odd, even	91
Cardinal FO	Less than eight, more than seven	92
Proportional	Less than half, more than half	85

Table 3
Mean (*M*) and standard deviation (*SD*) of the reaction time in milliseconds for each type of quantifier

Group	Quantifiers	*M*	*SD*
Aristotelian FO	All, some	2,257	471.95
Parity	Even, odd	5,751	1240.41
Cardinal FO	Less than eight, more than seven	6,035	1071.89
Proportional	Less than half, more than half	7,273	1410.48

sphericity, Greenhouse–Geiser adjustment was applied. Moreover, polynomial contrast analysis was performed for the within-subject factor.

Table 3 presents mean (*M*) and standard deviation (*SD*) of the reaction time in milliseconds for each type of quantifier.

We observed that the increase in reaction time was determined by the quantifier type ($F(2.4,94.3) = 341.24$, $p < .001$, $\eta^2 = .90$). Pairwise comparisons among means indicated that all four types of quantifiers differed significantly from one another ($p < 0.05$). Polynomial contrast analysis showed the best fit for a linear trend ($F(1,39) = 580.77$, $p < .001$). The mean reaction time increased as follows: Aristotelian quantifiers, parity quantifiers, cardinal quantifiers, proportional quantifiers.

4. Discussion

4.1. Conclusions

We have been studying verification of natural language quantifiers from the perspective of simple, automata-theoretic computational models. Our investigation is a continuation of previous studies. In particular, it enriches and explains some data obtained by McMillan et al. (2005) and Troiani et al. (2009) with respect to reaction times. Our results support the following conclusions:

4.1.1. Plausibility of the model

The automata-theoretic model correctly predicts that quantifiers computable by finite-automata are easier to understand than quantifiers recognized by push-down automata. It improves results of McMillan et al. (2005), which compared only first-order quantifiers with higher-order quantifiers, putting in one group quantifiers recognized by finite-automata and those recognized by push-down automata.

4.1.2. Aristotelian, cardinal, and parity quantifiers

We have observed a significant difference in reaction time between Aristotelian and parity quantifiers, even though they are both recognized by finite automata. This difference may be accounted for by observing that the class of Aristotelian quantifiers is recognized by acyclic finite automata, whereas in the case of parity quantifiers we need loops. Therefore, loops are another example of a computational resource influencing the complexity of cognitive tasks. Moreover, we have shown that processing first-order cardinal quantifiers of high rank takes more time than comprehension of parity quantifiers. This suggests that the number of states in the relevant automaton plays an important role when judging the difficulty of a natural language construction. Arguably, the number of states required influences hardness more than the necessity of using cycles in the computation. These observations shed some light on the differences between numerical and logical quantifiers assessed by Troiani et al. (2009).

4.1.3. Cognition and complexity

Last but not least, our research provides direct evidence for the claim that human linguistic abilities are constrained by computational resources (internal memory, number of states, loops).

4.2. Perspectives

There are many questions we leave for further research. Below we list a few of them.

4.2.1. Comprehension and brain

Even though we believe that computational properties are directly responsible for quantifier difficulty in natural language we are aware that our experiment does not support automata-theoretic account uniquely. However, our experimental setting can be used for neuropsychological study extending the one by McMillan et al. (2005). On the basis of our research and findings of McMillan et al. (2005), we predict that comprehension of parity quantifiers—but not first-order quantifiers—depends on executive resources that are mediated by dorsolateral prefrontal cortex. This would correspond to the difference between acyclic finite automata and finite automata. Further studies would contribute to extending our understanding of simple quantifier comprehension on Marr's implementation level.

4.2.2. Comprehension strategies

What about Marr's algorithmic level of explanation? It would be good to describe procedures actually used by our subjects to deal with comprehension. In principle it is possible to try to extract real algorithms by letting subjects manipulate the elements, tracking their behavior, and then drawing some conclusions about their strategies. This is one of the possible future directions to enrich our experiments.

4.2.3. Comprehension and working memory

Before starting any neuropsychological experiments it would be useful to measure memory involvement for different types of quantifiers using some classical methods known from cognitive psychology, like a dual-task paradigm combining a memory span measure with a concurrent processing task.

4.2.4. Comprehension beyond quantifiers

Finally, the automata-theoretic model can be extended for other notions than simple quantifiers. For example, as it was already suggested by van Benthem (1987), by considering richer data structures it can account for conditionals, comparatives, compound expressions in natural language, nonelementary combinations of quantifiers, link to learnability theory (see, e.g., Gierasimczuk, 2007), and others. Possibly such extensions could be valuable not only from a linguistic but also a cognitive point of view.

Acknowledgments

We would like to thank Johan van Benthem, Victor Ferreira, Nina Gierasimczuk, Theo Janssen, Marcin Mostowski, Jonathan Shaheen, and Iris van Rooij for comments on the previous version of the paper.

References

Baddeley, A. (2007). *Working memory, thought, and action. Oxford psychology series* (1st ed.). New York: Oxford University Press.

van Benthem, J. (1986). *Essays in logical semantics*. New York: Reidel.

van Benthem, J. (1987). Towards a computational semantics. In P. Gärdenfors (Ed.), *Generalized quantifiers* (pp. 31–71). London: Reidel Publishing Company

Cherniak, C. (1981). Minimal rationality. *Mind*, *90*(358), 161–183.

Church, A. (1936). An unsolvable problem of elementary number theory. *American Journal of Mathematics*, *58*(2), 345–363.

Clark, R., & Grossman, M. (2007). Number sense and quantifier interpretation. *Topoi*, *26*(1), 51–62.

Frege, G. (1892). Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, *100*, 25–50.

Frixione, M. (2001). Tractable competence. *Minds and Machines*, *11*(3), 379–397.

Gierasimczuk, N. (2007). The problem of learning the semantics of quantifiers. In B. Ten Cate & H. Zeevat (Eds.), *Lecture notes in computer science* (Vol. 4363, pp. 117–126). Berlin: Springer.

Hofstadter, D. (2007). *Gödel, Escher, Bach* (3rd ed.). New York: Tusquets.

Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2000). *Introduction to automata theory, languages, and computation* (2nd ed.). New York: Addison Wesley.

Lambalgen, M. V. & Hamm, F. (2005). *The proper treatment of events. Explorations in semantics*. Malden, MA: Wiley-Blackwell.

Levesque, H. J. (1988). Logic and the complexity of reasoning. *Journal of Philosophical Logic*, *17*(4), 355–389.

Marr, D. (1983). *Vision: A computational investigation into the human representation and processing visual information*. San Francisco: W.H. Freeman.

McMillan, C., Clark, R., Moore, P., Devita, C., & Grossman, M. (2005). Neural basis for generalized quantifiers comprehension. *Neuropsychologia*, *43*, 1729–1737.

McMillan, C., Clark, R., Moore, P., & Grossman, M. (2006). Quantifiers comprehension in corticobasal degeneration. *Brain and Cognition*, *65*, 250–260.

Mostowski, M. (1998). Computational semantics for monadic quantifiers. *Journal of Applied Non-Classical Logics*, *8*, 107–121.

Mostowski, M. & Wojtyniak, D. (2004). Computational complexity of the semantics of some natural language constructions. *Annals of Pure and Applied Logic*, *127*(1–3), 219–227.

van Rooij, I. (2008). The tractable cognition thesis. *Cognitive Science: A Multidisciplinary Journal*, *32*(6), 939–984.

Sanford, A. J., Moxey, L. M., & Paterson, K. (1994). Psychological studies of quantifiers. *Journal of Semantics*, *11*(3), 153–170.

Suppes, P. (1982). Variable-free semantics with remark on procedural extensions. In T. W. Simon & R. J. Scholes (Eds.), *Language, mind and brain* (pp. 21–34). Hillsdale: Erlbaum.

Szymanik, J. (2007). A comment on a neuroimaging study of natural language quantifier comprehension. *Neuropsychologia*, *45*(9), 2158–2160.

Szymanik, J. (2009). *Quantifiers in TIME and SPACE. Computational complexity of generalized quantifiers in natural language*. PhD thesis, Universiteit van Amsterdam.

Troiani, V., Peelle, J., Clark, R., & Grossman, M. (2009). Is it logical to count on quantifiers? Dissociable neural networks underlying numerical and logical quantifiers. *Neuropsychologia*, *47*(1), 104–111.

Tsotsos, J. (1990). Analyzing vision at the complexity level. *Behavioral and Brain Sciences*, *13*(3), 423–469.

Turing, A. (1936). On computable numbers, with an application to the Entscheidungs problem. *Proceedings of the London Mathematical Society*, *42*(2), 230–265.