Comprehension of Simple Quantifiers
Empirical Evaluation of a Computational Model

Jakub Szymanik and Marcin Zajenkowski
University of Amsterdam
University of Warsaw

November 6, 2008

Abstract

We compare time needed for understanding different types of quantifiers. In the first study, we show that the distinction between quantifiers recognized by finite-automata and push-down automata is psychologically relevant. In the second study, we compare comprehension of push-down quantifiers in universes with randomly placed objects and those where objects were ordered in some specific way simplifying (with respect to memory resources) computational task. The reaction time in the second case is significantly shorter than in the first case.

Keywords: language comprehension; working memory; generalized quantifiers; finite- and push-down automata; computational semantics of natural language

1 Introduction

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.

David Marr (1981) 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);

(3) implementation level (how it is actually done in neural activity).

Marr argued that the best way to achieve progress in cognitive science is by studying descriptions at computational level in psychological theories. He claims:

> An algorithm is likely to be understood more readily by understanding the nature of the problem being solved than by examining the mechanism (and the hardware) in which it is embodied. (Marr, 1981, p. 27)

Cognitive science has put a lot of effort into investigating the computational level of linguistic competence (see e.g. Isac and Reiss, 2008; Sun, 2008). Today computational restrictions are taken very seriously when discussing cognitive capacities. For instance, a psychological version of the Church-Turing thesis (Turing, 1936; Church, 1936) — stating that the human mind can only deal with computable problems — is commonly accepted. Moreover, complexity restrictions on cognitive tasks have already been noted in the philosophy of language and mind (see e.g. Cherniak, 1981; Chalmers, 1994; Hofstadter, 2007), the theory of language (see e.g. Mostowski and Wojtyniak, 2004; Levesque, 1988) and psychology of vision (see e.g. Tsotsos, 1990) leading to many variants of Tractable Cognition Thesis stating that human cognitive (linguistic) capacities are constrained by computational resources, like time and memory (see e.g. Frixione, 2001; van Rooij, 2008).

Unfortunately, there are not many empirical studies directly linking complexity predictions of computational models with psychological reality. The present research aims at increasing our empirical evidence in favor of this connection.

In the paper we are concerned with a very basic linguistic ability of understanding 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 (see Section 1.2) is psychologically plausible with respect to reaction time predictions.

Our research was motivated — among others things — by a recent neuropsychological investigation of the same problem (see McMillan et al., 2005) and, accordingly, by some troubles with the interpretation of its results (see Szymanik, 2007a). We discuss these matters in the next section. Then we provide readers with some mathematical details of an automata-theoretic model of quantifiers processing. In Sections 2 and 3 we present our empirical studies of some predictions that can be drawn from that model. We end with a summary and an outline of the future work.

1.1 Previous Investigations in the Area

Quantifiers have been widely treated from the perspective of cognitive psychology (see e.g. Sanford et al., 1994). However, research presented by McMillan et al. (2005) was the first attempt to investigate the neural basis of natural language quantifiers (see also McMillan et al. (2006) for evidence on quantifier comprehension in patients with focal neurodegenerative disease, and Clark and Grossman (2007) for more general discussion). It was devoted to study brain activity during comprehension of sentences with quantifiers. Using neuroimaging methods (BOLD fMRI) 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 (see e. g. van Benthem, 1986). One of the authors of the present paper has challenged this statement by invoking the computational difference between first-order quantifiers and divisibility quantifiers (see Szymanik, 2007a). 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 definable 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 definable 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".

The expressive power of higher-order quantifiers is much greater than the expressibility of first-order quantifiers. For instance, we cannot speak about infinite sets in first-order logic, but this is possible using higher-order quantifiers. This difference in expressive power corresponds to a difference in the computational resources required to check the truth-value of a sentence with those quantifiers.

In particular, to recognize first-order quantifiers we only need computability models which 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) (see e.g. Baddeley, 2007, for a psychological model).

(1) Every sentence in this paper is 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 (see Figure 2 for an illustration of a relevant automaton). We can proceed in a similar way for other first-order quantifiers. Formally, it was proved by Johan van Benthem (1986) that first-order quantifiers can be computed by such simple devices as finite automata (see Theorem 1 in

Section 1.2 containing mathematical details of the correspondence between quantifiers and automata).

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 (2) 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 correct.

Mathematically speaking, such an algorithm can be realized by a push-down automaton.

From this perspective, 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 computational complexity differences between first-order and higher-order quantifiers are also reflected in brain activity during processing quantifier sentences (McMillan et al., 2005, p. 1730). This hypothesis 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 quantifiers, which corresponds — with respect to memory resources — to the same computational model as first-order quantifiers.

McMillan et al. (2005) suggest that their study honours a distinction in complexity between classes of first-order and higher-order quantifiers. They also claim that:

> higher-order quantifiers can only be simulated by a more com-

plex computing device — a push-down automaton — which is equipped with a simple working memory device. (McMillan et al., 2005, p. 1730)

Unfortunately, this is not true. In fact, most of the quantifiers identified in the research as higher-order quantifiers can be recognized by finite automata. As we will see in the next chapter 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*.

1.2 Quantifiers and Automata

In what follows we give a short description of relevant mathematical results. We assume familiarity with basic terminology of automata theory and mathematical linguistics (see e.g. Hopcroft et al., 2000).

1.2.1 Simple Generalized Quantifiers

Generalized quantifiers are one of the basic tools of today’s linguistics and their mathematical properties have been extensively studied since the fifties (see Peters and Westerståhl, 2006, for a recent overview). In its simplest form — the one we are concerned with in this paper — Generalized Quantifier Theory assigns meaning to statements by defining semantics for quantifiers occurring in them. For instance, for quantifiers “every”, “some”, “at least 7”, “an even number of”, and “most” build up the following sentences.

(3) Every poet has low self-esteem.

(4) Some dean danced nude on the table.

(5) At least 7 grad students prepared presentations.

(6) An even number of the students saw a ghost.

(7) Most of the students think they are smart.

(8) Less than half of the students received good marks.

What is the semantics assigned to these quantifiers? Formally they are treated as relations between subsets of the universe. For instance, in sentence (3) "every" is a binary relation beetween set of poets and set of people having low self-esteem. Following the natural linguistic intuition we will say that sentence (3) is true if and only if the set of poets is included in the set of people having low self-esteem. Hence, the quantifier "every" corresponds in this sense to the inclusion relation. Let us now have a look at sentence (6). It is true if and only if the intersection of the set of all students with the set of people who saw a ghost is of even cardinality. Then this quantifier says something about parity of the intersection of two sets. Finally, let us consider example (7). We understand the quantifier "most" here as "more than half". Hence, sentence (7) is true if and only if the cardinality of the set of students who think they are smart is greater than cardinality of the set of students who do not think they are smart. Therefore, the quantifier "most" expresses that there is a specific proportion between these two kinds of students.

Formally the notion of a generalized quantifier was introduced by Andrzej Mostowski (1957) and developed further by Per Lindström (1966). It was introduced to linguistics by Barwise and Cooper (1981). Below we give a formal definition of monadic generalized quantifiers binding two unary variables:

Definition 1. *A monadic generalized quantifier binding two unary variables is a class* Q *of models of the form* $\mathbb{M} = (M, A, B)$*, where* A *and* B *are subsets of the universe* M*. Additionally,* Q *is closed under isomorphism.*

Let us explain this definition further by giving a few examples. First of all, models are precise mathematical representations of possible situations. Therefore, a class of models represents a class of possible situations, in this case corresponding to a quantifier.

Sentence (3) is of the form Every A *is* B, where A stands for poets and B for people having low self-esteem. As we explained above the sentence is true if and only if $A \subseteq B$. Therefore, according to the definition, the quantifier "every" corresponds to the class of models in which $A \subseteq B$. Because of the same reasons the quantifier "an even number of" corresponds to the class of models in which the cardinality of $A \cap B$ is an even number. Finally, let us consider the quantifier "most". As we mentioned before the sentence Most *As are B* is true if and only if $card(A \cap B) > card(A - B)$ and therefore the quantifier corresponds to the class of models where the inequality holds.

Why do we assume that these classes are closed under isomorphism? Simply put, it guarantees that the quantifiers are topic neutral. The quantifier "most" means exactly the same when applied to people as when applied to natural numbers.

Before we move to the computational model let us try to explain a distinction between first-order and higher-order quantifiers. Some generalized quantifiers, like "at least 3", "exactly 5", and "at most 8" are easily expressible in elementary logic. It is also true for many natural language determiners. For example, we can express the quantifier "some" by a first-order existential quantifier in the following way:

$$\mathsf{Some}\ x\ (A(x), B(x)) \iff \exists x (A(x) \wedge B(x)).$$

However, it is well-known that many generalized quantifiers are not definable in first-order logic. For example: "most", "every other", "odd", and "less than half" (see e.g. Peters and Westerståhl, 2006).

1.2.2 Representation of Finite Models

Having a quantified sentence and a model we would like to know how to compute the truth-value of this sentence in that model. The first step is to represent finite situations (models) as strings over some finite alphabet. In other words, we need to encode our finite models in a linear form. Here is the idea for doing it.

We restrict ourselves to finite models of the form $\mathbb{M} = (M, A, B)$. For instance, let us consider the model from Figure 1. We list all elements of the model in some order, e.g., $c_1, \ldots, c_5$. Then we replace every element in that sequence with one of the symbols from alphabet $\Gamma = \{a_{\bar{A}\bar{B}}, a_{A\bar{B}}, a_{\bar{A}B}, a_{AB}\}$, according to the constituents to which it belongs to. This means that we put the string of letters $a_{\bar{A}\bar{B}}$ in place of element c_1 as it belongs to the complement of set A (denoted as $\bar{A}$) and to the complement of set B. We write $a_{A\bar{B}}$ for element c_2 because it belongs to set A and to the complement of set B, and so on. As a result, in our example, we get the word $\alpha_M = a_{\bar{A}\bar{B}}a_{A\bar{B}}a_{AB}a_{\bar{A}B}a_{\bar{A}B}$. The word α_M corresponds to the model in which: $c_1 \in \bar{A}\bar{B}, c_2 \in A\bar{B} c_3 \in AB, c_4 \in \bar{A}B, c_5 \in \bar{A}B$. Hence, it uniquely (up to isomorphism) describes the model from Figure 1.

Definition 2. *The class* Q *corresponding to a quantifier is represented by the set of words (language)* $L_{\mathsf{Q}} \subseteq \Gamma^*$ *describing all elements (models) of the class.*

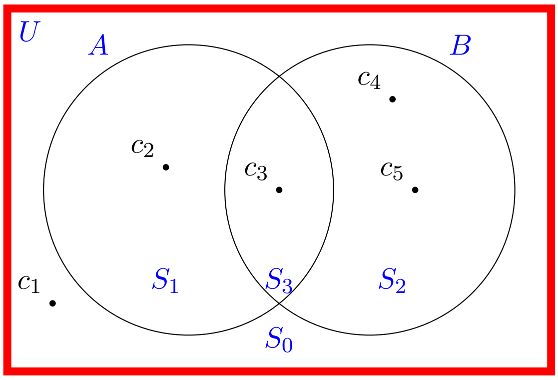

Figure 1: This model is uniquely described by $\alpha_M = a_{\bar{A}\bar{B}}a_{A\bar{B}}a_{AB}a_{\bar{A}B}a_{\bar{A}B}$.

1.2.3 Quantifier Automata

Having these definitions we would like to know what kind of automata correspond to particular quantifiers.

Aristotelian quantifiers Aristotelian quantifiers, like "all", "every", "some", "no", and "not all", are first-order definable. They need finite-automata with a fixed-number of state to be recognized. Let us consider an example.

Every A *is* B is true if and only if $A \subseteq B$. In other words, the sentence is true as long as there is no element belonging to A but not B. Having representation α_M of a finite model M over alphabet Γ we can easily recognize whether M satisfies sentence Every A *is* B. The following finite automaton from Figure 2 does the job. The automaton gets α_M as its input. It in-

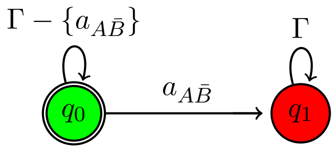

Figure 2: Finite automaton recognizing L_{Every}

spects the word letter by letter starting in the accepting state. As long as it does not find letter $a_{A\bar{B}}$ it stays in the accepting state, because it means that there was no element belonging to A but not to B. If it finds such an element (letter), then it already "knows" that the sentence is false and move to the rejecting state, where it stays no matter what happens next.

In other words, the quantifier "every" corresponds to the following regular language:

$$L_{\mathsf{Every}} = \{\alpha \in \Gamma^* : \#a_{A\bar{B}}(\alpha) = 0\},$$

where $\#c(\alpha)$ is the number of occurrences of the letter c in the word α.

Cardinal Quantifiers Cardinal quantifiers, e. g., "at least 3", "at most 7", and "between 8 and 11", like Aristotelian quantifiers are also first-order definable. However, the number of states of finite-automata recognizing cardinal quantifiers increases in proportion to the number that needs to be represented. Consider for example the following automata for At least three *As are B*:

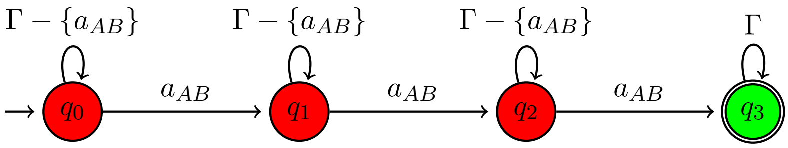

Figure 3: Finite automaton recognizing $L_{\mathsf{At\ least\ three}}$

This automata needs 4 states and it corresponds to the language:

$$L_{\mathsf{At\ least\ three}} = \{\alpha \in \Gamma^* : \#a_{AB}(\alpha) \geq 3\}.$$

Further, to recognize "at least 8" we would need 9 states and so on.

Divisibility Quantifiers What about the quantifier "an even number of"? It corresponds to the following regular language:

$$L_{\mathsf{Even}} = \{\alpha \in \Gamma^* : \#a_{AB}(\alpha) \text{ is even } \}.$$

The finite automaton from Figure 4 checks whether the number of occurrences of the letter a_{AB} in the string coding a given model is of even parity. It needs to remember whether it is in the "even state" (q_0) or the "odd state" (q_1) and loops between these states.

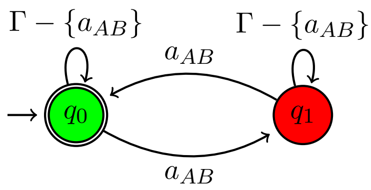

Figure 4: Finite automaton recognizing L_{Even}

Proportional Quantifiers Finally, let us have a look at the quantifier "most". The sentence Most *As are B* is true if and only if $card(A \cap B) > card(A - B)$. Therefore, the quantifier corresponds to the following context-free language:

$$L_{\mathsf{Most}} = \{\alpha \in \Gamma^* : \#a_{AB}(\alpha) > \#a_{A\bar{B}}(\alpha)\}.$$

There is no finite automata recognizing all such languages. As models might be of arbitrary finite cardinality so also the length of the coding strings is unbounded. In such a case it is impossible to compute "most" having only a fixed finite number of states as we are not able to predict how many states are needed. To give a computational device for this problem, some kind of internal memory, which allows the automaton to compare any number of occurrences of the symbols a_{AB} and $a_{A\bar{B}}$, is needed. A Push-down automata is a computational model that can achieve this by implementing the idea of a stack.

Characterization Above considered examples already give us a flavor of what is going on. Below we give a general answer to the question about computing devices recognizing particular quantifiers. We start by saying what it means that a class of monadic quantifiers is recognized by a class of devices.

Definition 3. *Let $\mathcal{D}$ be a class of recognizing devices, Ω a class of monadic quantifiers. We say that $\mathcal{D}$ accepts Ω if and only if for every monadic quantifier Q:*

$$\mathsf{Q} \in \Omega \iff \textit{there is a device } A \in \mathcal{D} \textit{ such that } A \textit{ accepts } L_{\mathsf{Q}}.$$

Now we are ready to state the relevant results. Quantifiers definable in first-order logic, FO, can be recognized by acyclic finite automata, which are a proper subclass of the class of all finite automata (van Benthem, 1986).

Theorem 1. *A quantifier Q is first-order definable iff L_{Q} is accepted by an acyclic finite automaton.*

A less known result due to Marcin Mostowski (1998) says that exactly the quantifiers definable in divisibility logic, $FO(\mathsf{D_n})$ (i.e. first-order logic enriched by all quantifiers "divisible by n", for $n \geq 2$), are recognized by finite automata (FA) .

Theorem 2. *A monadic quantifier Q is definable in the divisibility logic iff L_{Q} is accepted by a finite automaton.*

For instance, the quantifier $\mathsf{D_2}$ can be used to express the natural language quantifier "an even number of". An example of a quantifier falling outside the scope of divisibility logic is "most". Hence, it cannot be recognized by a finite-automaton[1].

Some quantifiers not definable in divisibility logic, like "most" and "less than half", can be recognized by push-down automata. Indeed, as we have

[1]More results on push-down automata and quantifiers can be found in a survey by Mostowski (1998), where a class of quantifiers recognized by the deterministic push-down automata is characterized. This class seems particularly interesting from the cognitive point of view.

argued, “most” is recognized by a push-down automaton. Obviously, the semantics of many natural language quantifier expressions cannot be modeled by such simple devices as PDA (Sevenster, 2006; Szymanik, 2007b).

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 definability properties, as those are not enough fine grained. A more careful 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 described in this section.

definability	examples	recognized by
FO	“all cars”, “some students", “at least 3 balls”	acyclic FA
$FO(D_n)$	“an even number of balls”	FA
not $FO(D_n)$	“most lawyers”, “less than half of the students”	PDA

Table 1: Quantifiers, definability, and complexity of automata.

1.3 The Present Experiment

Our experiment consists of 2 separate studies. The first study, described in Section 2, 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 and not only definability considerations. Additionally, we compare two classes of quantifiers inside the first-order group: Aristotelian and cardinal quantifiers.

The second study described in Section 3 dwells more on the engagement of working memory capacity in quantifier comprehension by using ordered and random distributions of the objects in pictures presented to participants.

1.3.1 General Idea of the 1st Study: Comparing Quantifiers

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

We predict that reaction time will increase along with the computational power needed to recognize quantifiers. Hence, divisibility quantifiers (even, odd) will take less 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 8". In the study of McMillan et al. (2005) only one cardinal quantifier of relatively small rank was taken into consideration, namely "at least 3". 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. Additionally, we suggest that 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 divisibility and proportional quantifiers.

1.3.2 General Idea of the 2nd Study: Quantifiers and Ordering

In the first study, sentences with pictures were presented to subjects, who had to decide whether the sentence was true. Array elements were randomly generated. However, the ordering of elements can be treated as an additional independent variable in investigating the role of working memory capacity. For example, consider the following sentence:

(9) Most As are B.

Although checking the truth-value of sentence (9) over an arbitrary universe needs a use of working memory, if the elements of a universe are ordered in pairs (a, b) such that $a \in A$, $b \in B$, then we can easily check it without using working memory. It suffices to go through the universe and check whether there exists an element a not paired with any b. This can be done by a finite automaton.

We have compared reaction time while subjects are judging the truth-value of statements containing proportional quantifiers, like sentence (9), over ordered and arbitrary universes. We predict that when dealing with an ordered universe working memory is not activated as opposed to when the elements are placed in an arbitrary way. As a result reaction time over ordered universes should be much shorter.

2 The First Study: Comparing Quantifiers

2.1 Participants

Forty native Polish-speaking adults took part in this study. They were volunteers from the University of Warsaw undergraduate population. 19 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 subject was tested individually and was given a small financial reward for participation in the study.

2.2 Materials and Procedure

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

(10) Some cars are red.

(11) 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 fifteen objects in two colors (see Figure 5).

Figure 5: An example of a stimulus used in the first study.

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 divisibility quantifiers (odd, even); the third was first-order cardinal quantifiers of relatively high rank (less than 8, more than 7); 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 8”, ”more than 7”, ”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 bee aware that each picture consisted of fifteen objects.

The experiment was divided into two parts: a short practice session followed immediately by the experimental session. Each quantifier problem was given one 15.5 s event. In the event the proposition and a stimulus array containing 15 randomly distributed cars were presented for 15000 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 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.

2.3 Results

2.3.1 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.

Quantifier group	Examples	Percent
Aristotelian FO	all, some	99
Divisibility	odd, even	91
Cardinal FO	less than 8, more than 7	92
Proportional	less than half, more than half	85

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

2.3.2 Comparison of Reaction Times

To examine the differences in means we used repeated measures analysis of variance with type of quantifier (4 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 sphericity, Greenhouse-Geiser adjustment was applied. Moreover, polynomial contrast analysis was performed for the within-subject factor. SPSS 14 was used for the analysis.

Table 3 presents 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	2257.50	471.95
Divisibility	even, odd	5751.66	1240.41
Cardinal FO	less than 8, more than 7	6035.55	1071.89
Proportional	less than half, more than half	7273.46	1410.48

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

We found out that the increase in reaction time was determined by the quantifier type ($F(2.4, 94.3) = 341.24$, $p < 0.001$, η^2=0.90). 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 ($F(1, 39) = 580.77$, $p < 0.001$). The mean reaction time increased as follows: Aristotelian quantifiers, divisibility quantifiers, cardinal quantifiers, proportional quantifiers (see Figure 6).

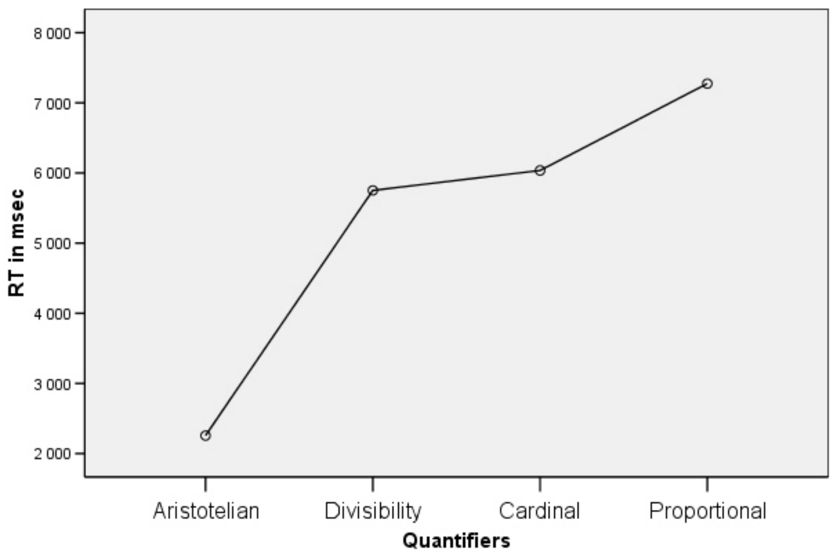

Figure 6: Average reaction times in each type of quantifiers in the first study.

3 The Second Study: Quantifiers and Ordering

3.1 Participants

Thirty native Polish-speaking adults took part in the second study. They were undergraduate students from two Warsaw universities. 12 were male and 18 were female. The mean age was 23.4 years (SD = 2.51) with a range of 20–28 years. Each subject was tested individually.

3.2 Materials and Procedure

In the task, we used sixteen grammatically simple propositions in Polish containing proportional quantifiers that probed a color feature of cars on a display (e.g. "More than half of the cars are blue"). Color pictures presenting a car park with 11 cars were constructed to accompany the propositions. As in the first study, the colors used for the cars were: red, blue, green, yellow, purple and black. Each picture contained objects in two colors.

Two different proportional quantifiers (less than half, more than half) were presented to each subject in 8 trials. Each type of sentence matched the picture in half of the trials. Moreover, each quantifier was accompanied with four pictures presenting cars ordered in two rows with respect to their colors (see Figure 7) and four pictures presenting two rows of randomly distributed cars. The rest of the procedure was the same as in the first study.

3.3 Results

3.3.1 Analysis of Accuracy

The behavioral data showed higher accuracy of subjects' judgments for ordered universes (89% correct) than for unordered universes (79% correct).

3.3.2 Comparison of Reaction Times

Since there were only two types of situations (random and ordered) in the study, a paired-samples t-test was used to analyze differences in the reaction times. Proportional quantifiers over randomized universes (M=6185.93;

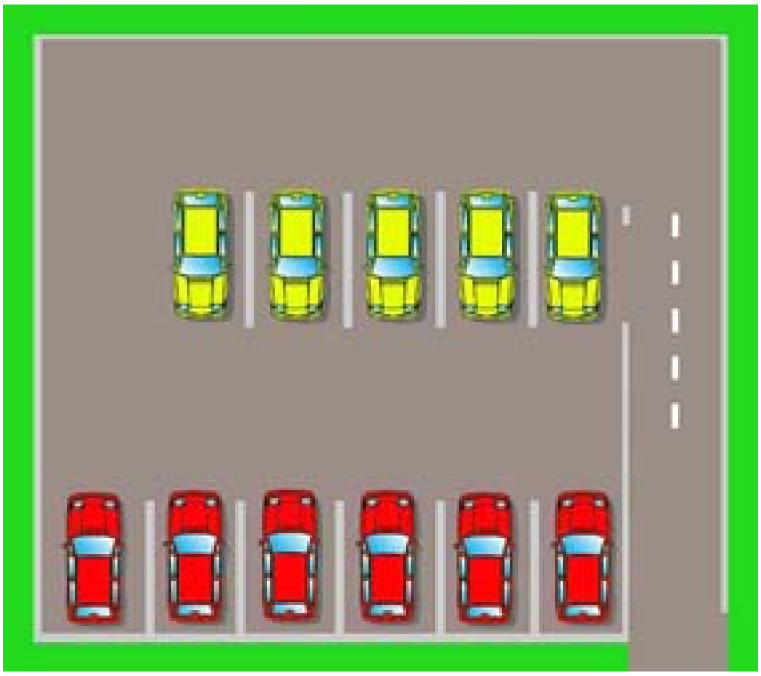

Figure 7: An example of a stimulus used in the second study. A case when cars are ordered.

SD=1759.09) were processed significantly longer than these over ordered models (M=4239.00; SD=1578.26) ($t(29) = 5.87$ $p < 0.001$; $d = 1.16$).

4 Conclusions and Perspectives

We have been studying comprehension 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) with respect to reaction times. Our results support the following conclusions:

(1) The automata-theoretic model described in Section 1.2 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.

(2) We have observed a significant difference in reaction time between Aristotelian and divisible 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 divisible quantifiers we need loops. Therefore, loops are another example of computational resources having influence on the complexity of cognitive tasks.

(3) 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.

(4) Decreased reaction time in the case of proportional quantifiers over ordered universes supports findings of McMillan et al. (2005), who attributed the hardness of these quantifiers to the necessity of using working memory.

(5) 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).

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

(1) 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 divisibility 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. Moreover, we expect that only

quantifiers recognized by PDAs but not FAs activate working memory (inferior frontal cortex). Additionally, the inferior frontal cortex should not be activated when judging the truth-value of sentences with proportional quantifiers over ordered universes. Further studies would contribute to extending our understanding of simple quantifier comprehension on Marr's implementation level.

(2) What about the 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.

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

(4) We find it interesting to explore differences in comprehension of Aristotelian and cardinal quantifiers in more detail, both from the empirical and theoretical points of view. It would provide us with an opportunity to better understand the connection between the number of states as a part of computational models and the real cognitive capacities described by these models.

(5) It has been observed by Geurts (2003) that monotonicity plays a crucial role in reasoning with quantifiers (see also Geurts and van der Silk, 2005). Upward monotone quantifiers are easier than downward monotone ones[2] with respect to reasoning. It is the matter of empirical testing to check whether the same holds for comprehension. Our study was not designed to explore this possibility. However, we compaired

[2]A quantifier Q is upward monotone (increasing) iff the following holds: if $\mathsf{Q}(A, B)$ and moreover $B \subseteq B'$ then $Q(A, B')$. The downward monotone (decreasing) quantifiers are defined analogously as being closed on taking subsets.

pairs of quantifiers with respect to monotonicity in the right argument and observed the following. In the case of the Aristotelian quantifiers ”all“ and ”some“ monotonicity influences reaction time for comprehension in a way close to being significant. Divisibility quantifiers are non-monotone, but we have observed that ”odd“ is more difficult. For cardinal first-order quantifiers we have a significant result: the decreasing quantifier ”less than 8“ is more difficult than its increasing counterpart. Unfortunately, we did not observe any statistical dependencies between proportional quantifiers of different monotonicity. We leave investigation of the role of monotonicity in comprehension for a future research.

(6) Finally, the automata-theoretic model can be extend 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, non-elementary combinations of quantifiers, link to lernability theory (see e.g. Gierasimczuk, 2007) and others. Possibly such extensions could be valuable not only from linguistic but also cognitive point of view.

Acknowledgments

The first author was supported by a Marie Curie Early Stage Research fellowship in the project GLoRiClass (MEST-CT-2005-020841). We would like to thank Johan van Benthem, Nina Gierasimczuk, Theo Janssen, and Jonathan Shaheen for comments on the previous version of the paper.

References

Baddeley, A. (2007). *Working Memory, Thought, and Action*. Oxford Psychology Series. Oxford University Press, USA, 1st edition.

Barwise, J. and Cooper, R. (1981). Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4:159–219.

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

van Benthem, J. (1987). Towards a computational semantics. In Gärdenfors, P., editor, *Generalized Quantifiers*, pages 31–71. Reidel Publishing Company.

Chalmers, D. (1994). A computational foundation for the study of cognition. See: http://consc.net/papers/computation.html.

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

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

Geurts, B. (2003). Reasoning with quantifiers. *Cognition*, 86(3):223–251.

Geurts, B. and van der Silk, F. (2005). Monotonicity and processing load. *Journal of Semantics*, 22:97–117.

Gierasimczuk, N. (2007). The problem of learning the semantics of quantifiers. In Ten Cate, B. and Zeevat, H., editors, *Logic, Language, and Computation, 6th International Tbilisi Symposium on Logic, Language, and Computation, TbiLLC 2005*, volume 4363 of *Lecture Notes in Computer Science*, pages 117–126, Batumi, Georgia. Springer.

Hofstadter, D. (2007). *Gödel, Escher, Bach.* Tusquets, 3rd edition.

Hopcroft, J. E., Motwani, R., and Ullman, J. D. (2000). *Introduction to Automata Theory, Languages, and Computation.* Addison Wesley, 2nd edition.

Isac, D. and Reiss, C. (2008). *I-Language: An Introduction to Linguistics as Cognitive Science.* Oxford University Press, USA.

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

Lindström, P. (1966). First order predicate logic with generalized quantifiers. *Theoria*, 32:186–195.

Marr, D. (1981). *Vision: A Computational Investigation into the Human Representation and Processing Visual Information.* W.H. Freeman, San Francisco.

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

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

Mostowski, A. (1957). On a generalization of quantifiers. *Fundamenta Mathematicae*, 44:12–36.

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

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

Peters, S. and Westerståhl, D. (2006). *Quantifiers in Language and Logic.* Clarendon Press, Oxford.

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

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

Sevenster, M. (2006). *Branches of imperfect information: logic, games, and computation.* PhD thesis, Universiteit van Amsterdam.

Sun (2008). *The Cambridge Handbook of Computational Psychology.* Cambridge University Press, 1st edition.

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

Szymanik, J. (2007b). A strong meaning hypothesis from a computational perspective. In Aloni, M., Dekker, P., and Roelofsen, F., editors, *Proceedings of the Sixteenth Amsterdam Colloquium*, pages 211–216. University of Amsterdam.

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 Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(2):230–265.