Quantifiers in TIME and SPACE : computational complexity of generalized quantifiers in natural language

Szymanik, J.K.

Publication date
2009

Chapter 7

Comprehension of Simple Quantifiers

In this chapter we step back from complex quantifiers to consider a simpler case. We are interested here in quantifiers binding two unary variable (monadic quantifiers of type (1, 1)). As we will see their meanings can be explained in terms of finite-state and push-down automata. Therefore, from the computational complexity perspective we are very low in the hierarchy of computational problems (see Section 2.3.3).

Restricting ourselves to tractable natural language constructions we will be able to directly argue for the relevance of computational complexity theory for cognitive science. Actually, we will empirically show that more complex quantifiers are more difficult to comprehend. What do we mean by that?

In this chapter we compare the time needed for understanding different types of quantifier. Our results are consistent with an automata-theoretic model of processing monadic quantifiers in natural language which has been posited by several linguists and logicians. We show that the distinction between quantifiers recognized by finite automata and push-down automata is psychologically relevant. Moreover, our results point out the influence of computational resources on the complexity of cognitive tasks as discussed in Chapter 1. Additionally, our research clarifies the recent neuropsychological research on quantifier comprehension. In particular, we directly consider the computational properties of quantifiers and not their logical definability. We also throw more light on the involvement of working memory by comparing comprehension of quantifiers over discourse universes with randomly placed objects and those where objects are ordered in some specific way, simplifying the computational task with respect to memory resources.

This chapter is based on the paper (Szymanik, 2007a) and joint work with Marcin Zajenkowski (see Szymanik and Zajenkowski, 2008).

7.1 Motivations

One of the primary objectives of cognitive sciences is to explain human information processing. As we have mentioned in Chapter 1 we are mainly concerned with the computational level (see Marr, 1983) of this task. 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 (see Chapter 1 for more discussion). Unfortunately, there are not many empirical studies directly linking the complexity predictions of computational models with psychological reality. The present research aims at increasing our empirical evidence in favor of this connection.

We are concerned here with the very basic linguistic ability of understanding sentences, and consistently identify meaning with a computing procedure (see Chapter 1). In particular, we are dealing here 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 in pictures. We show that a simple computational model describing the processing of such sentences — presented in Section 7.1.2 — is psychologically plausible with respect to reaction time predictions.

Our research was motivated — among other things — by a recent neuropsychological investigation into the same problem and, accordingly, by some problems with the interpretation of its results. We discuss these matters in the next section before we provide readers with some mathematical details of the automata-theoretic model of quantifier processing we are working with. Sections 7.2 and 7.3 present our empirical studies of some predictions that can be drawn from that model. We end with a summary and an outline of future work. The basic notions of mathematical linguistics and automata theory we are using can be found in Section 2.3.1 of the Prerequisites chapter.

7.1.1 Previous Investigations in the Area

Quantifiers have been widely treated from the perspective of cognitive psychology (see e.g. Sanford et al., 1994). However, the 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); Troiani et al. (2009) for a more general discussion). This paper was devoted to a study of 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 lan-

guage quantifier comprehension posited by several linguists and logicians (see e.g. van Benthem, 1986). We have challenged this statement by invoking the computational difference between first-order quantifiers and divisibility quantifiers (see Szymanik, 2007a). The starting point of the recent 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. 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) All sentences in this chapter are correct.

It suffices to read the sentences from this chapter one by one. If we find an incorrect one, then we know that the statement is false. Otherwise, if we read the entire chapter without finding any incorrect sentence, then statement (1) is true (see Figure 7.2 for an illustration of a relevant automaton). We can proceed in a similar way for other first-order quantifiers. Formally, it has been proved by van Benthem (1986) that all first-order quantifiers can be computed by such simple devices as finite automata (see Theorem 7.1.8 in Section 7.1.2, which contains the 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 chapter are correct.

Mathematically speaking, such an algorithm can be realized by a push-down automaton (see Theorem 7.1.10 in the next section).

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 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 complex 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 section 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*.

7.1.2 Monadic Quantifiers and Automata

In what follows we give a short description of the relevant mathematical results. We assume familiarity with the basic terminology of automata theory which we surveyed in Section 2.3.1.

Monadic Generalized Quantifiers

In this chapter we are concerned with generalized quantifier theory restricted to its simplest form. We want to assign a meaning for sentences like the following:

(3) All poets have 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.

We have explained in detail the semantics assigned to these quantifiers in Section 2.2, where we also formally defined the notion of a generalized quantifier. Below we give a formal definition of monadic generalized quantifiers of type (1, 1). These are the quantifiers we are working with in this chapter.

7.1.1. DEFINITION. A monadic generalized quantifier of type (1, 1) 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 isomorphisms. ■

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 7.1. 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. 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 the set A (denoted as $\bar{A}$) and to the complement of the set B. We write $a_{A\bar{B}}$ for element c_2 because it belongs to the set A and to the complement of the 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 7.1.

7.1.2. DEFINITION. The class Q corresponding to a quantifier is represented by the set of words (language) L_{Q} describing all elements (models) of the class. ■

We can extend this idea of coding to generalized quantifiers of any type (see e.g. Mostowski, 1998). For a monadic quantifier binding n variables we will in general need an alphabet consisting of 2^n letters, one letter for every constituent. However, if we restrict ourselves to so-called CE-quantifiers of type (1, 1), i.e., satisfying isomorphism closure, extension (domain independence) and conservativity

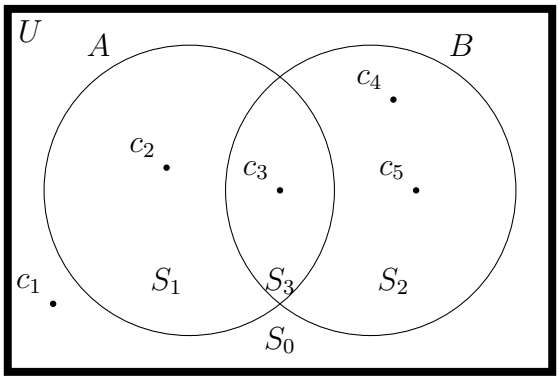

Figure 7.1: This model is uniquely described by $\alpha_M = a_{\bar{A}\bar{B}}a_{A\bar{B}}a_{AB}a_{\bar{A}B}a_{\bar{A}B}$.

(see Section 2.2.5) — which arguably cover all determiners in natural language — then we would need only two letters, one for the elements belonging to the first argument but not the second one and the other for the elements in the intersection of the arguments. This observation follows from Theorem 2.2.19. It suggests that this coding is a quite psychologically plausible representation in case of CE-quantifiers.

Quantifier Automata

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

Aristotelian Quantifiers The Aristotelian quantifiers "all", "some", "no", and "not all", are first-order definable. They need finite automata with a fixed number of states. Let us consider an example.

7.1.3. Example. All *As are 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 All *As are B*. The following finite automaton from Figure 7.2 does the job. The automaton gets α_M as its input. It inspects the

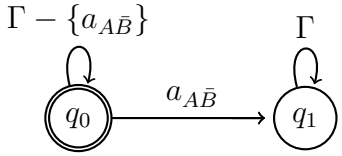

Figure 7.2: Finite automaton recognizing L_{All}

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 this 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 moves to the rejecting state, where it stays no matter what happens next.

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

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

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 the Aristotelian quantifiers are also first-order definable. However, the number of states of a finite automaton recognizing a cardinal quantifier increases in proportion to the number that needs to be represented.

7.1.4. Example. Consider for example the following automaton for $\mathsf{At\ least\ three}\ A\text{s are } B$:

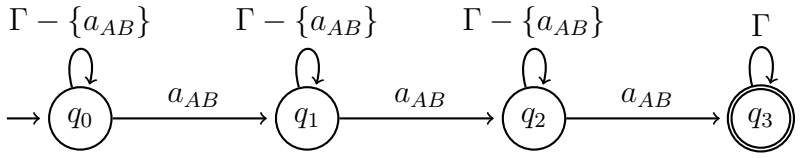

Figure 7.3: Finite automaton recognizing $L_{\mathsf{At\ least\ three}}$

This automaton needs four states and it corresponds to the language:

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

Furthermore, to recognize "at least 8" we would need nine states and so on.

Parity Quantifiers

7.1.5. Example. 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 7.4 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.

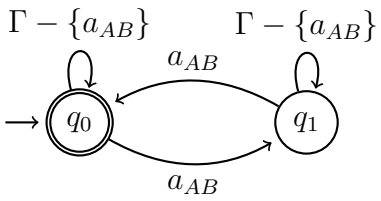

Figure 7.4: Finite automaton recognizing L_{Even}

Proportional Quantifiers

7.1.6. Example. Finally, let us have a look at the quantifier "most". The sentence Most *As are B* is true if and only if $\mathrm{card}(A \cap B) > \mathrm{card}(A - B)$. 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 automaton 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 (see Section 2.3.1 for mathematical justification). 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 (the particular push-down automaton needed for recognizing L_{Most} is similar to the automaton for the language L described in Section 2.3.1).

Characterization The examples considered above already give us the 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.

7.1.7. Definition. 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:

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

■

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).

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

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

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

For instance, the quantifier 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.

A partial characterization of the quantifiers recognized by push-down automata is also known. For example, quantifiers of type (1) expressible in the arithmetic of addition (additive quantifiers, semi-linear quantifiers), so-called Presburger Arithmetic (PrA), are recognized by push-down automata (PDA) (van Benthem, 1986).

7.1.10. THEOREM. *A quantifier* Q *of type (1) is definable in PrA iff* L_{Q} *is accepted by a push-down automaton.*

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

Obviously, the semantics of many natural language quantifier expressions cannot be modeled by such a simple device as a PDA. Just think about sentence (9) whose meaning cannot be computed by any push-down automaton (it corresponds to the language L_{abc} from Section 2.3.1).[1] This fact follows from the Pumping Lemma for context-free languages (see Theorem 2.3.11).

(9) An equal number of logicians, philosophers, and linguists climbed K2.

There are of course much more complex expressions in natural language, like those discussed in the previous sections of the thesis.

To sum up, first-order and higher-order quantifiers do not always differ with respect to 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 fine-grained enough. A more careful perspective — taking into account all the results we've mentioned, which are summed up in Table 7.1 — will 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.

[1] At least without assuming any additional invariance properties for the quantifier in question.

Definability	Examples	Recognized by
FO	"all cars", "some students", "at least 3 balls"	acyclic FA
FO(D_n)	"an even number of balls"	FA
PrA	"most lawyers", "less than half of the students"	PDA

Table 7.1: Quantifiers, definability, and complexity of automata.

7.1.3 The Present Experiment

Our experiment consists of two separate studies. The first study, described in Section 7.2, compares the 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 the predictions of 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 7.3 dwells more on the engagement of working memory in quantifier comprehension by using ordered and random distributions of the objects in pictures presented to participants.

General Idea of the First Study: Comparing Quantifiers

First, we compared reaction time with respect to the following classes of quantifiers: those recognized by an acyclic FA (first-order), those recognized by an FA (parity), and those recognized by a PDA. 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) (see Definition 4.3.3).

Moreover, we have additionally compared the 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 the complexity of the mental processing of cardinal quantifiers depends on the number of states in the relevant automaton. Therefore, cardinal quantifiers of high rank should be more difficult than Aristotelian quantifiers. Additionally, we suggest that the number of states in an automaton (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.

General Idea of the Second Study: Quantifiers and Ordering

There are many possible ways of verifying the role of working memory capacity in natural language quantifier processing. One way we have taken into account is as follows.

In the first study, sentences with pictures were presented to subjects, who had to decide whether the sentence was true. The 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. For example, consider the following sentence:

(10) Most As are B.

Although checking the truth-value of sentence (10) 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 times while subjects are judging the truth-value of statements containing proportional quantifiers, like sentence (10), 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.

7.2 The First Study: Comparing Quantifiers

7.2.1 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.

7.2.2 Materials and Procedure

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

(11) Some cars are red.

(12) 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 7.5).

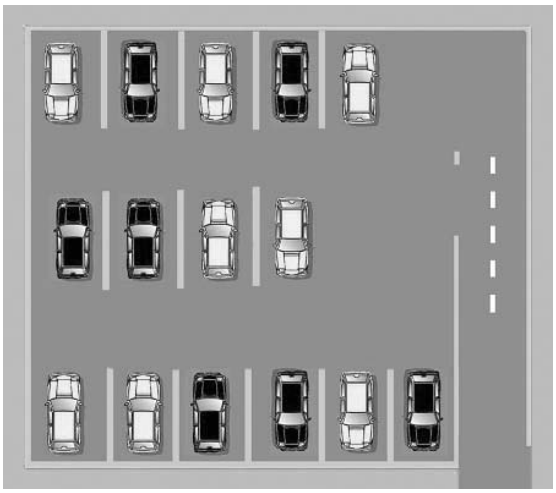

Figure 7.5: An example of a stimulus used in the first study.

Eight different quantifiers divided into four groups were used in the study. The first group of quantifiers were first-order Aristotelian quantifiers (all, some); the second were parity quantifiers (odd, even); the third were first-order cardinal quantifiers of relatively high rank (less than 8, more than 7); and the fourth were proportional quantifiers (less than half, more than half) (see Table 7.2). 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 target objects and fifteen in total) for "less than half"). Debriefing following the experiment revealed that none of the participants had been 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 of the presented picture. They responded by pressing the key "P" if true and the key "F" if false. The letters refer to the first letters of the Polish words for "true" and "false".

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

7.2.3 Results

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 7.2.

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

Table 7.2: The percentage of correct answers for each group of quantifiers.

Comparison of Reaction Times

To examine the differences in means we used a 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 7.3 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
Parity	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 7.3: 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, parity quantifiers, cardinal quantifiers, proportional quantifiers (see Figure 7.6).

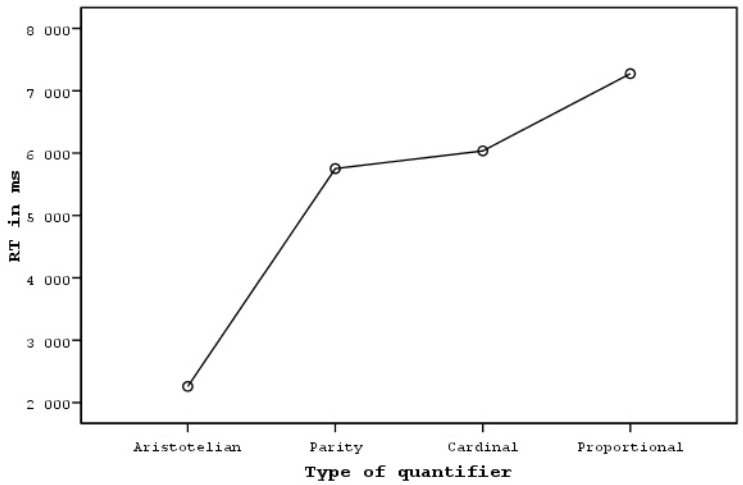

Figure 7.6: Average reaction times in each type of quantifiers in the first study.

7.3 The Second Study: Quantifiers and Ordering

7.3.1 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.

7.3.2 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 eleven 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 by four pictures presenting cars ordered in two rows with respect to their colors (see Figure 7.7) and four pictures presenting two rows of randomly distributed cars. The rest of the procedure was the same as in the first study.

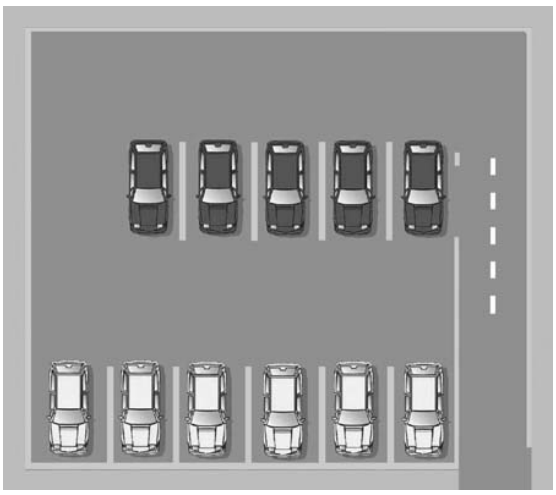

Figure 7.7: An example of a stimulus used in the second study. A case when cars are ordered.

7.3.3 Results

Analysis of Accuracy

The behavioral data showed higher accuracy of subjects' judgments for ordered universes (90% correct) than for unordered universes (79% correct) (see Table 7.4).

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

Table 7.4 summarizes the results of this study.

Situation	Accuracy	M	SD
Randomized	79%	6185.93	1759.09
Ordered	90%	4239.000	1578.27

Table 7.4: Accuracy, mean (M) and standard deviation (SD) of the reaction time in milliseconds for proportional quantifiers over randomized and ordered universes.

7.4 Summary

Conclusions

We have been studying the 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:

- The automata-theoretic model described in Section 7.1.2 correctly predicts that quantifiers computable by finite automata are easier to understand than quantifiers recognized by push-down automata. It improves the 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.

- 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 a computational resource having an influence on the complexity of cognitive tasks.

- 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.

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

- 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).

Perspectives

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

7.4.1. QUESTION. Our experimental setting can be used for neuropsychological studies extending the one by McMillan et al. (2005). On the basis of our research and the 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 the 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. Are these predictions correct? Further studies answering this question would contribute to extending our understanding of simple quantifier comprehension on Marr's implementation level.

7.4.2. QUESTION. What about the algorithmic level of explanation? It would be good to describe the 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.

7.4.3. QUESTION. 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. Will these methods confirm working memory engagement according to the predictions?

7.4.4. QUESTION. We find it interesting to explore the differences in comprehension of Aristotelian and cardinal quantifiers in more detail, both from the empirical and theoretical points of view. This 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. How does the number of states in the minimal automaton influence the difficulty of the quantifier?

7.4.5. QUESTION. 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 with respect to reasoning. It is a matter of empirical testing to check whether the same holds for comprehension. Our study was not designed to explore this possibility. However, we compaired 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. Parity 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. What is the role of monotonicity in quantifier comprehension?

7.4.6. QUESTION. Finally, the automata-theoretic model can be extended for other notions than simple quantifiers. For example — as was already suggested by van Benthem (1987) — by considering richer data structures it can account for conditionals, comparatives, compound expressions in natural language, and non-elementary combinations of quantifiers (like branching); also it can form a link with learnability theory (see e.g. Gierasimczuk, 2007) and others. Are such possible extensions of any value for cognitive science?