ELSEVIER

Neuropsychologia 45 (2007) 2158–2160

NEUROPSYCHOLOGIA

www.elsevier.com/locate/neuropsychologia

Current controversies

A comment on a neuroimaging study of natural language quantifier comprehension

Jakub Szymanik [1]

Institute for Logic, Language, and Computation, University of Amsterdam, Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands
Institute of Philosophy, Warsaw University, Poland

Received 6 December 2006; received in revised form 17 January 2007; accepted 21 January 2007
Available online 30 January 2007

1. Neuroimaging data

Research presented in this journal by McMillan, Clark, Moore, Devita, and Grossman (2005) is the first attempt to investigate the neural basis of natural language quantifiers (see also McMillan, Clark, Moore, and Grossman (2006) for evidence on quantifier comprehension in patients with focal neurodegenerative disease and Clark and Grossman (2006) for more general discussion). It was devoted to study brain activity during comprehension of sentences with generalized quantifiers. Using 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 (e.g. see van Benthem, 1986). I challenge this statement by invoking the computational difference between first-order quantifiers and divisibility quantifiers (e.g. see Mostowski, 1998). Moreover, I suggest other studies on quantifier comprehension, which can throw more light on the role of working memory in processing quantifiers.

1.1. First-order and higher-order quantifiers

The authors 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. 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 expressibility 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 the 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 working memory. Intuitively, to check whether sentence (1) is true we do not have to remember anything.

(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 statement (1) 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 proved by van Benthem (1986) that first-order quantifiers can be computed by such simple devices as finite automata.

However, for recognizing some higher-order quantifiers, like “less than half” or “most”, we need computability models making use of working 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.

E-mail address: szymanik@science.uva.nl.
[1] The author is a recipient of the 2006 Foundation for Polish Science Grant for Young Scientists.

0028-3932/$ – see front matter
doi:10.1016/j.neuropsychologia.2007.01.016

From this perspective, the authors hypothesized that all quantifiers recruit the right inferior parietal cortex, which is associated with numerosity. Taking the distinction about 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 anatomy during processing quantifier sentences (McMillan et al., 2005, p. 1730). This hypothesis was confirmed.

2. Discussion

In my view, the authors' interpretation of their results is not convincing. Also, 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 working memory – to exactly the same computational model as first-order quantifiers. Let us have a closer look at the paper of McMillan et al. (2005).

2.1. Quantifiers and working memory

The authors 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 completely true. Most of the quantifiers qualified in the research as higher-order quantifiers can be recognized by finite automata. Both “an even number” and “an odd number” are quantifiers recognizable by two-state finite automata with transition from the first state to the second and *vice versa*. 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 (3) is true you do not need to count the number of incorrect sentences and then compare it with that of the set of even integers.

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

You need only remember parity. For example when you find an incorrect sentence you write “1” at the blackboard, if you find another one you erase “1” and put “0” again, then if you see another incorrect 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.

Table 1
Quantifiers and complexity of corresponding algorithms

Definability	Example	Recognized by
FO	“all cars”, “some students”, “at least 3 balls”	Acyclic FA
FO(D_n)	“an even number of balls”	FA
Pr	“most lawyers”, “less than half of the students”	PDA

In what follows we give a short description of relevant mathematical 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). A less known result due to Mostowski (1998) says that 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). For instance, quantifier D_2 can be use to express the natural language quantifier “an even number of”.

Quantifiers of type (1) not definable in FO (D_n) but expressible in the arithmetic of addition, so-called Presburger arithmetic (PR), are recognized by push-down automata (PDA) (van Benthem, 1986). Push-down automata are computability models making essential use of working memory in the form of a so-called stack. Obviously, semantics of many natural language quantifier expressions cannot be modeled by such simple devices as PDA (Table 1).

My criticism is that first-order and higher-order quantifiers do not differ with respect to working memory requirement. Therefore, the explanation of brain activation patterns proposed by the authors is based on the wrong assumption. A simple automata-theoretic perspective is not enough to describe the processing of natural language quantifiers. Some additional arguments need to be found for interpreting the results. In what follows I will propose a few ways of exploring the subject empirically.

3. Improving experiment

3.1. First-order and divisibility quantifiers

We should compare brain activation with respect to the three classes of quantifiers: recognizable by acyclic FA (first-order), FA (divisibility), and PDA. I do not know whether the authors compared these classes. If they did, then it would be important to analyze it. However, the authors did not report any data on these differences.

Specifically, I predict differences between first-order and divisibility quantifiers. Comprehension of divisibility quantifiers – but not first-order quantifiers – should depend on executive resources that are mediated by dorsolateral prefrontal cortex. It would correspond then to the difference between acyclic finite automata and finite automata.

We expect that only quantifiers not definable in divisibility logic will activate working memory (inferior frontal cortex).

3.2. Aristotelean and cardinal quantifiers

It would be also interesting to compare Aristotelean quantifiers, like "all", "every", "some", "no", "not all", with cardinal quantifiers, e.g. "at least 3", "at most 7", "between 8 and 11". They are all definable in first-order logic, but elementary representation of cardinal quantifiers can be ill-suited for psychological purposes. Consider, for example how "at least 3 balls" is translated into first-order logic:

$$\exists x \exists y \exists z (x \neq y \wedge y \neq z \wedge x \neq z \wedge ball(x) \wedge ball(y) \wedge ball(z)).$$

Since we cannot talk about sets in elementary logic, then – as you can deduce from the above example – the complexity of first-order translation of cardinality quantifiers is proportional to the rank of the cardinal that needs to be represented.

In the reported study, only one cardinal quantifier of relatively small rank was taken into consideration, namely "at least 3". It might be the case that the mental processing complexity of cardinal quantifiers is more similar to that of higher-order quantifiers than to Aristotelean. However, to observe this, cardinal quantifiers of higher rank should be used, for instance "at least 7". Obviously, this issue is strongly connected with the phenomena of subitizing as opposed to counting.

3.3. Quantifiers and ordering

Finally, there are many possible ways of verifying the role of working memory in natural language quantifier processing. One way is as follows. In the reported research, subjects were presented sentences with visual arrays and had to decide whether a sentence was true. Array elements were randomly generated. However, ordering of elements can be treated as an additional independent variable to investigate the role of working memory. For example, consider the following sentence:

(4) Most As are B.

Although checking the truth-value of sentence (4) over an arbitrary universe needs use of a kind 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. It would be interesting to carefully compare the pattern of neuroanatomic recruitment while subjects are judging the truth-value of statements, like sentence (4), over ordered and arbitrary universes. We predict that when dealing with ordered universe working memory will not be activated, but it will be if the elements are placed in arbitrary way.

Acknowledgements

The author would like to express his appreciation to Theo Janssen for comments on the manuscript. This research was supported by a Marie Curie Early Stage Research fellowship in the project GloRiClass (MEST-CT-2005-020841).

References

Clark, R., & Grossman, M. (2006). Number sense and quantifier interpretation, *Topoi*, submitted for publication.

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. *Brian and Cognition*, *65*, 250–260.

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

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