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

Szymanik, J.K.

Publication date
2009

Chapter 8

Conclusions and Perspectives

8.1 General Summary

In the thesis we have pursued the topic of the computational complexity of natural language quantifier constructions. Our perspective combined logical research with linguistic insights and an empirical, cognitive foundation.

In Chapter 1 we discussed computational semantics as a part of the Dynamic Turn in the philosophy of language. In particular, we gave a short history and argued for identifying a basic element of meaning — confronting it with the actual world — with algorithms. Assuming the procedural theory of meaning allows us to view natural language comprehension in a broader perspective of cognitive science. Moreover, it suggests that we should look for theoretical constraints in computational complexity theory. Aiming for that general perspective we recalled the three levels of explanation in cognitive psychology proposed by Marr: computational, algorithmic, and neurological. Next, we noticed that studying the computational complexity of natural language constructions can contribute to the computational level of explanation. Among other possibilities, as computational complexity theory deals with abstract, inherent, and hardware-independent properties of information processing tasks it can be used for explaining the cognitive difficulty of human performance. Following the literature, we proposed treating problems computable in polynomial time as easy (tractable) and problems not computable in polynomial time (NP-hard) as difficult (intractable). Our technical work, in the following chapters, was mostly devoted to drawing tractability distinctions between the meanings of various natural language quantifier constructions, and evaluating the plausibility of the proposal.

Additionally, assuming the distinction between tractable and intractable and following Ristad (1993), we have argued in Section 1.8 that a good semantic theory of the everyday fragment of natural language should be expressible in the existential fragment of second-order logic. Simply put, its descriptive power should be high enough to account for the flexibility of everyday language, but, on the other hand, a theory can not be too strong or it will overgeneralize the linguistic data and need to postulate implausible linguistic mechanisms.

Let us briefly review how our technical results throw light on these grand issues identified in Chapter 1.

In Chapters 3 and 5 we studied the logical and computational structures of existing linguistic theories and obtained some new complexity results. In Chapter 3 we studied polyadic quantifiers in natural language. We showed that the most common operations creating polyadic quantifiers in everyday language — Boolean operations, iteration, cumulation, and resumption — do not lead outside the tractable class of semantic constructions. On the other hand, more sophisticated polyadic lifts playing an important role in linguistics — branching and Ramseyfication — can produce expressions with intractable referential meanings. In Chapters 4, 5, and 6 we investigated fragments of natural language semantics, drawing some linguistic conclusions from complexity observations. First of all, our results show that computational complexity might be a factor constraining the expressive power of everyday language. For example, it might be the case that collective quantification in natural language cannot really express all possible facts about collections. The interpretation process has to be restricted by some other linguistic factors to keep its complexity reasonable. On the other hand, we have shown that linguistic theories have to take computational complexity into account as long as they want to serve as an input for plausible cognitive theories.

Additionally, in Chapter 4 we studied some linguistic cases where model-checking is intractable and the inferential aspect of comprehension comes into the game. This was the case with the different readings of reciprocal expressions which are connected by inferential properties. Our results suggest that the two basic aspects of meaning, model-theoretic and inferential, as identified in Chapter 1, are strongly linked and that their interaction might be triggered by computational complexity effects. This shows how computational complexity can influence pragmatics.

Finally, in Chapter 7 we described an empirical study supporting the computational complexity perspective on meaning. These observations directly linked complexity to difficulty for natural language. They also show the strength of the procedural approach to semantics with respect to formulating interesting hypotheses of psychological relevance. This clear link between algorithmic semantics and cognitive science makes a dynamic approach to meaning extremely attractive and proves its relevance.

In general, our research has explored the advantages of identifying meaning with an algorithm. We have shown the fruitfulness of this approach for linguistics and cognitive science.

There are of course many related issues waiting for an explanation. We have discussed some open questions following directly from our work at the end of each chapter. Below we review some more general questions and directions for future research. This may help in forming a better picture of the research endeavor this dissertation is a part of.

8.2 Outline

8.2.1 Different Complexity Measures

Recall that in Chapter 3 we showed that proportional Ramsey quantifiers define NP-complete classes of finite models. On the other hand, we have also observed that bounded Ramsey quantifiers are in PTIME. It is an open problem where the precise border lies between tractable and mighty Ramsey quantifiers. As we already noted in Section 3.4 the above open problem directs us towards parametrized complexity theory. In general the following question arises:

8.2.1. Question. What is the parametrized complexity of iteration, cumulation, resumption, branching, and Ramseyfication (the operations studied in Chapter 3)?

Using parametrized complexity can help us to find the boundary between tractable and mighty Ramsey quantifiers. Moreover, it can answer some doubts about worst-case complexity as a measure of linguistic difficulty (see Section 1.5.3). We can already notice that also from the parametrized complexity perspective the clique problem is believed to be intractable. Therefore, our general claims about the intractability of strong reciprocity would probably be preserved under this measure.

Broadly speaking we would like to study the complexity of natural language semantics from various perspectives. Therefore, we ask:

8.2.2. Question. What is the complexity of quantifiers under different measures, like parametrized, circuit, and average-case complexity?

Sevenster (2006) has already noted that it would be interesting to characterize the circuit complexity of quantifiers. The other interesting measure would be average-case complexity. It could help to answer questions like the following: How difficult is it to compute quantifiers on average (random) graphs? Here the situation might be different than for worst-case complexity. For instance, the clique problem (the strong interpretation of reciprocal expressions) can be solved on average in sub-exponential time on random graphs. Therefore, from this perspective Ramsey quantifiers can become "almost" tractable in most cases.

8.2.2 Quantifiers and Games

We have motivated our interest in the computational complexity of quantifiers by the insight it gives into the possibilities for processing natural language determiners. However, besides studying various measures of complexity we can also investigate the properties of evaluation games for quantifiers. In the case of existential and universal quantifiers and their combinations such games have been extensively studied (see e.g. Hintikka and Sandu, 1997). However, the issue has never been satisfactory worked out for a wider class of generalized quantifiers, including polyadic constructions (see e.g. Clark, 2007).

8.2.3. Remark. Our suggestion for future work is to identify simple quantifiers with games and then investigate combinations of these games coinciding with polyadic lifts on simple quantifiers.

There is already some work in that direction. For instance, Peter Aczel (1975) formulated a game-theoretical interpretation for infinite strings of monotone quantifiers, more than 30 years ago. Recently, similar ideas were proposed by van Benthem (2002, 2003). He considered a so-called game logic which encodes the algebra of game operations. In particular, he has shown that the algebra of sequential operations, like choice, dual and composition, coincides with the evaluation games for first-order logic (see van Benthem, 2003). Recently, van Benthem et al. (2007) have extended this idea and proposed a complete logic, which they call Concurrent Dynamic Game Logic, to formalize simultaneous games. This logic — with the product operator representing parallel games — has applications in studying the branching operation. An immediate open question stemming from that research is as follows:

8.2.4. Question. Does concurrent dynamic logic coincide with evaluation games for first-order logic extended by all Henkin quantifiers (see van Benthem et al., 2007)?

Studying the game algebra corresponding to polyadic combination of quantifiers can help to understand the structure of natural operations creating complex games for compound quantifier expression out of simple ones. This fresh perspective might be valuable for solving some of the notoriously difficult problems of compositionality.

Moreover, games might be useful for formulating an intuitive semantics for second-order definable quantifiers in arbitrary weak models (see Section 1.6). In a finite universe second-order definable quantifiers correspond to alternating computing (see Theorem 2.4.5) which is similar to game-theoretical semantics. Maybe this idea can be naturally extended to the case of infinite weak models.

Additionally, one can think about using generalized quantifiers to define solution concepts in games. For example, we can express the existence of an extreme Nash equilibrium in a game using a first-order formula.

8.2.5. Question. What about branching quantifiers? Can they account for the existence of mixed equilibria in non-determined games? Maybe studying other generalized quantifiers in that context can lead to conceptually new game solutions. If we enrich first-order logic with generalized quantifiers we can count, or talk about proportions. Does this give rise to any interesting game concepts?

8.2.3 Cognitive Difficulty and Complexity

We need to investigate the interplay between cognitive difficulty and computational complexity in more detail. Do the differences in computational complexity really play an important role in natural language processing as our data from Chapter 7 suggests? Can we show empirically the influence of computational complexity on the difficulty of other cognitive tasks? For example:

8.2.6. Question. Can we design experiments confirming our speculations from Section 4.5, i.e., that shifts in reciprocal meaning are sometimes triggered by the computational complexity of sentences?

8.2.7. Question. Is there an empirical way to prove the implausibility of the higher-order approach to collective quantification studied in Chapter 5?

Moreover, it is very important to extend the complexity explanation beyond the first level of Marr's hierarchy (described in Section 1.5.1). We should not only study the abstract computational properties of natural language constructions but also try to grasp the procedures really used by humans to deal with the comprehension of natural language. It would in principle be possible to extract real strategies by letting subjects manipulate the elements, tracking their behavior and then drawing some conclusions about their strategies. This is one of the possible directions for enriching our experiments from Chapter 7. Obviously, the most ambitious question is to describe comprehension at the neural implementation level using a brain scanning technique. Hopefully, at some point there will be enough cognitive data to start broad fMRI studies of the problem.

8.2.4 Future of GQT and Beyond

Finally, we would like to share one general impression on the state of the art in generalized quantifier theory. Recently, Peters and Westerståhl (2006) have published an excellent monograph in the field. The book focuses on definability questions and their relevance for linguistics. It gives the impression that the research field is almost complete and there are not so many interesting open questions. In fact, one can observe decreasing interest in generalized quantifier theory since the eighties. On the other hand, the above-mentioned handbook contains no chapter that deals with computational complexity issues, collective quantification or cognitive science. It was already noticed in the review by van Benthem (2007) that "classical" generalized quantifier theory as presented in the book considers logic and language, but misses out on computation, which should be the third pillar.

We hope that our dissertation may be seen as laying the groundwork for strengthening generalized quantifier theory with the computational pillar, which might help to revive the field. We have tried to convey a general message that there is still a lot to do in the field of generalized quantifiers. We have mainly focused on complexity questions and their interplay with cognitive science but there are many more possibilities. For instance, the problem of investigating the invariance properties of collective quantifiers (second-order generalized quantifiers), as we have mentioned in Chapter 5, is not only very classical in spirit but also potentially would have a large impact on linguistics and theoretical computer science. Other examples of interesting research directions which can be taken in the context of generalized quantifiers include game-theoretical analysis, learnability theory for quantifiers and its connections with invariance properties like monotonicity.

Last but not least, it is necessary to extend the computational complexity analysis to different natural language constructions as well as quantifiers; for instance, exploiting the automata approach involving richer data structures, as was already proposed in the eighties (see e.g. van Benthem, 1987). Moreover, we have argued in the first chapter that it would be natural to treat dynamic theories of language, like belief-revision, theories of context-dependence, signalling games, and discourse representation theories, with a computational complexity analysis. Presumably we need a new model of computation which is better conceptual fit to the task of describing communication in language. One obvious candidate would be a model based on games, which have already been embraced by computer scientists as a rich model of computations.

Summing up, a fresh computational approach is needed to evaluate the cognitive plausibility of dynamic theories. And, if necessary, this can lead to more natural reformulations. Moreover, complexity analysis can be a first step towards connecting linguistic theories of communication with cognitive science. After all, the ability to use language is one of many human cognitive processes and as such should not be analyzed in isolation.