Design and Evaluation of Formal Representations: An Incremental Approach using Logic Grid Puzzles

Koen Hindriks* and Joost Vromans†

It has been suggested to use *logic puzzles* as a test suite for computational systems that are able to answer questions based on a precise understanding of the natural language input (see [2]). There are several benefits of using logic puzzles for evaluating natural language understanding systems. Most importantly, logic puzzles are created independently from the particular goals that are put on the agenda by researchers interested in natural language processing. To solve such puzzles, moreover, a computational system has to be able to handle a wide variety of linguistic phenomena and be able to reason with natural language input. Finally, logic puzzles define precise and well-structured problems, which set clear evaluation criteria for the performance of the computational system trying to solve such puzzles.

There is a variety of logic puzzles that one can use for such evaluation purposes. The example presented in [2] is drawn from a Law School Admission Test. The idea is that the main skills required for solving small, logical puzzles such as these consist of basic natural language understanding and logical reasoning. Apart from some very general knowledge about temporal and spatial constraints, additional common sense knowledge about the way the world is typically is not required. There are other types of logic puzzles, however, that are more demanding in this respect. The types of logic puzzles that we have in mind are so-called *logic grid puzzles*. These puzzles consist of a short introduction that sets the scenario and introduces the problem to be solved, and a series of premisses or *clues* which are sufficient to solve the problem raised. One of the most famous examples of such puzzles is the so-called *Zebra Puzzle*, which asks the question *Who Owned the Zebra?*, and which is also known as *Einstein's puzzle*.

Logic grid puzzles, we believe, can provide us with an effective test for the quality of a formal representation. To solve logic puzzles, a formal representation is needed in order to represent semantic structure as well as the

*Delft University of Technology, k.v.hindriks@tudelft.nl

†Radboud University Nijmegen, joostvromans@student.ru.nl

background knowledge. A precise understanding of text by a computational system will require a system that is able to map natural language input into the semantic representation and is able to reason with that input combined with any other available background knowledge. The areas of computational semantics and knowledge representation can both benefit from a test case such as logic grid puzzles, since both are concerned with representation and reasoning. The point is that successful natural language understanding systems presuppose the design of an appropriate representation that reflects our commonsense view of the world *and* the way we talk about it.

The design of such a representation is one of the main challenges facing computational semantics and knowledge representation. In this abstract, we discuss an incremental approach and method to develop such a representation. We believe this approach to computational semantics will provide a basis for effective and precise natural language understanding. From a practical point of view, we believe this research will eventually enable more sophisticated natural language database interfaces and techniques for extracting information from texts. From a theoretical point of view, it will provide insight into formal representations that are suitable for both natural language understanding as well as knowledge representation in general.

Logic Grid Puzzles as a Test Case Logic grid puzzles vary in difficulty and in the diversity of linguistic phenomena they present. As an example, a very simple logic grid puzzle is shown in Figure 1. It is not hard to derive the solution depicted in the grid from the clues provided. Even in this simple example, howevever, basic lexical knowledge about the relation between *love* and *like* is assumed since the question is posed as: *Can you figure out who likes which colour, and who has which age?* Many more puzzles, typically with larger grids, are available in puzzle magazines such as *Logic Problems*.

The Zebra Puzzle mentioned above is more challenging: Additional background knowledge is required because the clues leave out some details, for example that houses are lined up in a row. Moreover, since the puzzle does not mention a zebra in the clues, it must be inferred that someone owns a zebra. These types of problems make clear that a representation is needed that also can be used to handle implicit assumptions which are present in natural language texts. Designing a computational system that is aimed at the precise understanding of natural language, we believe, requires that such interaction between semantic and background knowledge is taken seriously.

There are three main reasons why we think that logic grid puzzles are particularly suitable for testing the quality of a formal representation. First, these puzzles pose real challenges even for simple puzzles. To solve more complicated puzzles, moreover, a computational system must be able to handle quantifiers and scope, comparatives, tense, pronouns and anaphoric references, relative clauses, different types of adverbs, ordinals, etc.

Second, logic grid puzzles provide a basis for an incremental approach that aims at the gradual refinement of the representation language. That is, one can start with first-order logic to translate very simple puzzles, and add in subsequent steps additional features that allow the handling of anaphora, or events. Third, these puzzles are very well-structured and always come with a unique solution. The solution to a logic grid puzzle simply *is* the (unique) model that satisfies the natural language (and some additional puzzle) constraints. This means that model checking tools can be used to find such a solution by translating the natural language clues into a formal representation. This problem formulation resembles the approach presented in [1]. This provides a very clear evaluation criterion for evaluating the adequacy of the formal representation used.

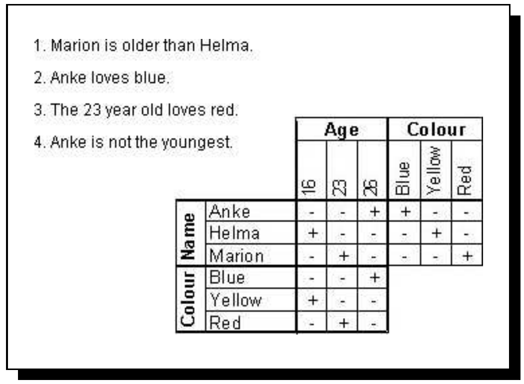

Fig. 1: Example Logic Grid Puzzle

Design & Evaluation of Formal Representations We briefly present a step-by-step process to develop and evaluate formal representations for natural language understanding using logic grid puzzles.

The process is cyclic and illustrated in Fig. 2. The challenges that logic grid puzzles pose to a computational system for natural language understanding require an *integrated* solution to different problems. Various semantic representations have been proposed to handle specific linguistic phenomena. An integrated computational approach provides a means to evaluate these proposals.

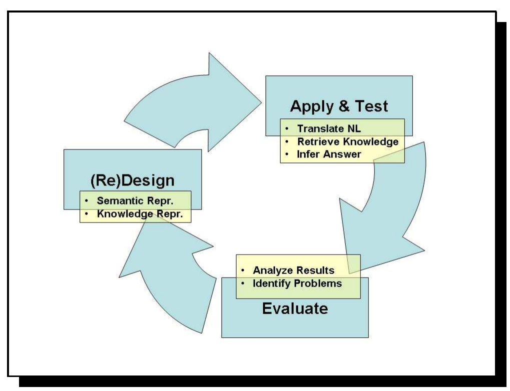

Fig. 2: Design, Apply & Test, Evaluate Cycle

The process consists of three steps (see figure 2):

1. Design: Initially, develop/select a formal representation based on an analysis of some simple puzzles to get started. Furthermore, a specific set of logic puzzles should be identified for testing. After several cycles, additional puzzles should be analyzed and closely scrutinized in order to improve the formal representation. Various linguistic phenomena may require additions to the formal representation.

2. Apply & Test: Apply the representation in a computational system that uses it to translate the logic puzzle clues and reasons with it to derive an answer. This involves a.o. implementing any required extensions to the grammar and knowledge modules.

3. Evaluate: Finally, analyze and evaluate the results. Identify whether problems occurred due to the representation, reasoning or knowledge components. Basic grammar problems will not be identified here, since proper handling of the syntax is assumed.

After one cycle, the process should be repeated. The scope of extensions, however, should be carefully determined at the start of every cycle. In particular, such extensions should be conversative and based on *empirical findings*: that is, extensions to the grammar and semantics should be motivated by phenomena present in the puzzles that are in scope. Using this approach, the main drivers for (re)designing the semantic/knowledge representation are complexity of the linguistic phenomena and problems that were identified in the evaluation step of the cycle.

Conclusion Computational semantics offers the prospect of a precise understanding of natural language by computers. One of the main challenges however is to find an appropriate formal representation that can be effectively used for translating linguistic phenomena as well as for reasoning. To evaluate the quality and enable the development of such a representation, we argue that logic grid puzzles provide a good starting point. Starting with the simplest logic puzzles, by means of an incremental step-by-step process of extending and evaluating formal representations eventually more complex natural language phenomena may be handled by a computational system.

References

[1] P. Blackburn and J. Bos. *Representation and Inference for Natural Language. A First Course in Computational Semantics.* CSLI, 2005.

[2] I. Lev, B. MacCartney, C.D. Manning, and R. Levy. Solving Logic Puzzles: From Robust Processing to Precise Semantics. In *Proceedings of the 2nd Workshop on Text Meaning and Interpretation (ACL'04)*, 2004.