Do we need the closed-world assumption in knowledge representation?

Ullrich Hustadt*
Max-Planck-Institut für Informatik
Im Stadtwald, D-66123 Saarbrücken
e-mail hustadt@mpi-sb.mpg.de

1 Introduction

Database systems and knowledge representation systems represent and reason about some aspect of the real world. In both it is common to separate the two functions of *representation*, i.e. describing the conceptual scheme and the actual data, and *computation*, i.e. answering of queries and manipulation of data.

The database management system of a database system provides a data definition language to describe the conceptual scheme. The data definition language is used to describe the database in terms of a data model. Operations on the database require a specialized language, called a data manipulation language or query language. One of the most important data models is the *relational model* which describes the world in terms of atomic values and relations on the set of all atomic values. Data manipulation languages of the relational model comprise the relational algebra, and the domain and tuple relational calculi. The *object-oriented model* supports a more elaborated description of the world by allowing complex objects, i.e. objects constructed using record formation and set formation, classes, i.e. abstract data types describing methods, which are operations to be performed on the objects, and class hierarchies.

The data manipulation languages of these data models are based on the following assumptions.

The closed-world assumption
which says that all information that is not true in the database is considered as false.

The unique-name assumption
which says that two distinct constants (either atomic values or objects) necessarily designate two different objects in the universe.

The domain-closure assumption
which says that there are no other objects in the universe than those designated by constants of the database.

These assumptions are important to understand the way computations are performed in databases.

Knowledge representation formalisms are aimed to represent general conceptual information and are typically used in the construction of the knowledge base of a reasoning agent. A knowledge base can be thought of as representing the beliefs of such an agent. One of the most prominent knowledge representation formalisms is KL-ONE [Brachman and Schmolze,1985] which has been used in the construction of natural language processing systems.

The knowledge representation language of KL-ONE and all it's derivates can be considered as a subset of first-order logic with equality. With respect to describing structural properties of objects and conceptual schemes they are more expressive than the data definition languages corresponding to the relational or object-oriented model.

In the late eighties inference in KL-ONE was shown to be undecidable [Schmidt-Schauss,1989]. Since then the emphasis in research has been on developing and investigating systems that are computationally well behaved, i.e. are tractable or at least decidable [Brachman *et al.*,1991; Donini *et al.*,1991; Buchheit *et al.*,1993]. As a result many commonly used knowledge representation languages have restricted expressiveness and are in their current form no longer suitable for natural language applications. They are still more expressive than data definition languages, but the question can be risen whether there is an application needing this additional expressive power.

Nevertheless, data manipulation languages and query languages of knowledge representation formalisms differ considerably in their underlying assumptions.

The open-world assumption
which says that there can be true facts that are not contained in the knowledge base.

The unique-name assumption
which says that two distinct constants (either atomic values or objects) necessarily designate two different objects in the universe.

The open-domain assumption
which says that there can be more objects in the universe than those designated by constants in the knowledge base unless a constraint in the knowledge base prevents this.

That means, that even if the data definition language and the data manipulation language of a database management system and a knowledge base management system would coincide, the results of data manipulations would differ.

***Acknowledgments:** This work has been supported by the German Ministry for Research and Technology (BMFT) under grant ITS 9102 (Project Logo). Responsibility for the contents lies with the author.

In the next section I will give some examples that show the usefulness of closed-world inferences in natural language processing. Thus knowledge representation languages sticking to the open-world assumption seem to be insufficient for natural language processing.

2 Query answering in Natural Language Processing

In cooperation with the PRACMA Project[1] (Department of Computer Science, University of Saarbrücken) we have been developing a suitably extended knowledge representation system, called MOTEL [Hustadt and Nonnengart,1993], which is intended to be a module of the PRACMA system. The PRACMA Project [Jameson *et al.*,1994] is concerned with the modeling of noncooperative information-providing dialogues. An example from PRACMA's domain is the dialogue between a person $\mathcal{S}$ trying to sell her used car to a potential buyer $\mathcal{B}$. Naturally, the goals of $\mathcal{S}$ conflict in part with those of $\mathcal{B}$.

In the final implementation, the natural language analysis module of the PRACMA system will use the semantic representation language $\mathcal{NLL}$ [Laubsch and Nerbonne,1991] to represent the German-language input strings. The resulting $\mathcal{NLL}$ expressions will be stored in the pragmatic dialogue memory. Various modules will process the content of the dialogue memory, the most important one for us is the *comment and question handler*. The result of this module is transfered to the natural language generator which is responsible for verbalizing $\mathcal{NLL}$ expressions.

$\mathcal{NLL}$ contains a first-order logic core with anadic predicates, generalized quantifiers, plural reference expressions, and λ-abstraction. To fit the purposes of PRACMA the language has been extended by modal operators.

Suppose the knowledge base of the car seller S contains declarations defining that vehicles are either cars or trucks, `veh1` is a truck, and `veh2` is a vehicle. This can be represented in $\mathcal{NLL}$ in the following way.

```
(forall ?x vehicle(inst: ?x) iff
           (car(inst: ?x) or
            truck(inst: ?x))                (1)
truck(inst: veh1)                           (2)
vehicle(inst: veh2)                         (3)
```

Here `veh1` and `veh2` are constants, `vehicle`, `car`, and `truck` are predicate symbols. In $\mathcal{NLL}$ arguments of predicates are identified via keywords, e.g. `inst`, rather than positions in argument vectors. Any identifier preceded by a question mark, e.g. `?x`, is a variable. In addition we have used the boolean operators `iff` (equivalence) and `or` (disjunction), and the universal quantifier `forall` in declaration (1).

Now a question of the buyer concerning which objects are either cars or trucks is represented in the following way.

```
(?lambda ?x car(inst: ?x) or
            truck(inst: ?x))                (4)
```

An expression of the `(?lambda ?x` P`)` denotes the set of all `?x` satisfying P. The answer we have to infer from the knowledge base is that `veh1` and `veh2` both belong to this set.

Obviously, this answer cannot be computed by the comment and question handler without taking declaration (1) into account. For instance, it is not possible to find the correct answer to (4) by computing the answer sets for `(?lambda ?x car(inst: ?x))` and `(?lambda ?x truck(inst: ?x))` and to return the union of the resulting sets as an answer.

A question of the buyer concerning which objects do not belong to the set of trucks is translated into the following $\mathcal{NLL}$ expression.

```
(?lambda ?x not car(inst: ?x))              (5)
```

Whereas the closed-world assumption would allow us to infer that `veh1` belongs to this set, the open-world assumption underlying $\mathcal{NLL}$ doesn't support this conclusion.

The question whether all cars are vehicles can also be formulated in $\mathcal{NLL}$. To answer this question we can try to infer

```
(forall ?x vehicle(inst: ?x) if
           car(inst: ?x))                   (6)
```

from the knowledge base. The answer to this question has to be independent of the constants currently occurring in our knowledge base. On the basis of declaration (1), the answer has to be positive.

Now let us assume that the left front seat of `veh2` is red. Choosing `lfseat` to designate the left front seat, this can be represented in the following way.

```
hasPart(inst: veh2, theme: lfseat)          (7)
seat(inst: lfseat)                          (8)
hasColour(inst: lfseat, theme: red)         (9)
```

To answer the question whether all seats of `veh2` are red we have to try to infer the following $\mathcal{NLL}$ expression.

```
(forall ?x
        hasColour(inst: ?x, theme: red)
        if hasPart(inst: veh2, theme: ?x)
           and seat(inst: ?x))              (10)
```

Because of the open-domain and open-world assumption, the answer to the question cannot be positive. Although the only seat the car seller knows to be part of `veh2` is actually red, there may be other seats of `veh2` and these seats may not be red.

Intuitively, a positive answer is much more plausible. We would assume that the car seller knows all the seats of `veh2` and knows the colour of every seat of `veh2`. It is possible to extend the knowledge base using *number restrictions* in such a way that we can infer a positive answer, e.g.

```
((= 1) ?x hasPart(inst: veh2, theme: ?x)
          and seat(inst: ?x))               (11)
```

[1] PRACMA is short for 'PRocessing Arguments between Controversially Minded Agents.'

declares that **veh2** has exactly one seat. declarations (7),(8),(9), and (11) taken together allow us to answer query (10) positively. However, it seems to be more natural to extend the language by an *epistemic modal operator* in the style of Lifschitz [Lifschitz,1991] to solve the problem. For a description of an extension of the knowledge representation language $\mathcal{ALC}$ by an epistemic operator refer to Donini et al. [Donini *et al.*,1992].

Suppose our language contains such an epistemic operator **K**. Then we have two possibilities to get a positive answer to the question. The first possibility is to reformulate the question slightly in the following way.

```
(forall ?x
    hasColour(inst: ?x, theme: red) if
    K(hasPart(inst: veh2, theme: ?x)
      and seat(inst: ?x)))                    (12)
```

Now the question is whether all known seats of **veh2** are red and the answer has to be positive. This approach causes the problem how the natural language analysis module should determine the epistemic character of question (12) opposed to the non-epistemic character of question (6).

The second possibility is to add a declaration of the following form to the knowledge base

```
not (hasPart(inst: veh2, theme: ?x)
     and seat(inst: ?x)) if
 not K(hasPart(inst: veh2, theme: ?x)
       and seat(inst: ?x))                    (13)
```

This declaration allows to conclude that an object is either not part of **veh2** or not a seat if it is not known to be part of **veh2** and a seat.

Obviously, we are now able to turn our knowledge base system into a database system either by suitably adding epistemic operators to all the queries or by adding enough epistemic rules to the knowledge base. Therefore, the extension of knowledge representation languages with an epistemic operator is a first step to unify the database world and the knowledge base world.

3 Future Work

It is well-known that theorem proving in a first-order language containing an epistemic operator is not even semi-decidable. Although the answers to the example questions presented in the previous section seem to be derived easily, there is no hope to find a correct and complete inference mechanism which is able to deduce them.

If we need a correct inference mechanism, the only possibility we have is to restrict the knowledge representation language, i.e. we have to identify a decidable fragment of $\mathcal{NLL}$ to which we can add an epistemic operator without loosing decidability.

References

[Brachman and Schmolze, 1985] Ron J. Brachman and J. G. Schmolze. An Overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

[Brachman *et al.*, 1991] Ron J. Brachman, Deborah L. McGuinness, Peter F. Patel-Schneider, and A. Borgida. Living with CLASSIC: When and how to use a KL-ONE-like language. In J. F. Sowa, editor, *Principles in Semantic Networks: Explorations in the Representation of Knowledge*, pages 401–456. Morgan Kaufmann, San Mateo, California, 1991.

[Buchheit *et al.*, 1993] M. Buchheit, F. M. Donini, and A. Schaerf. Decidable reasoning in terminological knowledge representation systems. Research Report RR-93-10, Deutsches Forschungszentrum für Künstliche Intelligenz, Saarbrücken, Germany, 1993.

[Donini *et al.*, 1991] F. M. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The complexity of concept languages. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, pages 151–162, Cambridge, USA, April 22–25 1991. Morgan Kaufmann.

[Donini *et al.*, 1992] F. M. Donini, M. Lenzerini, D. Nardi, A. Schaerf, and W. Nutt. Adding Epistemic Operators to Concept Languages. In B. Nebel, C. Rich, and W. Swartout, editors, *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, pages 342–353, Cambridge, USA, 1992. Morgan Kaufmann.

[Hustadt and Nonnengart, 1993] U. Hustadt and A. Nonnengart. Modalities in knowledge representation. In Chris Rowles, Huan Liu, and Norman Foo, editors, *Proceedings of the 6th Australian Joint Conference on Artificial Intelligence*, pages 249–254, Melbourne, Australia, 16–19 November 1993. World Scientific.

[Jameson *et al.*, 1994] Anthony Jameson, B. Kipper, A. Ndiaye, R. Schäfer, J. Simons, T. Weis, and D. Zimmermann. Cooperating to be noncooperative: The dialog system pracma. To appear in the Proceedings of the 18th Annual German Conference on Artificial Intelligence, 1994. Springer.

[Laubsch and Nerbonne, 1991] J. Laubsch and J. Nerbonne. An Overview of $\mathcal{NLL}$. Technical report, Hewlett Packard Laboratories, May 1991.

[Lifschitz, 1991] Vladimir Lifschitz. Nonmonotonic databases and epistemic queries. In *Proceedings of the Twelfth International Conference on Artificial Intelligence*, pages 381–386, Sydney, Australia, August 24–30 1991. Morgan Kaufmann.

[Schmidt-Schauss, 1989] M. Schmidt-Schauss. Subsumption in KL-ONE is Undecidable. In R. J. Brachman and H. J. Levesque, and R. Reiter, editors,*Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pages 421–431, Toronto, Canada, May 15–19 1989. Morgan Kaufmann.