Using HOL LIGHT to Reason over Second-Order MRLs

Michael Minock

TCS/CSC
KTH Royal Institute of Technology
minock@kth.se

Abstract

This extended abstract presents some very preliminary work exploring higher-order *meaning representation languages* (MRLs) for natural language interfaces over databases. Specifically the HOL LIGHT theorem prover is being applied to deduce query containment for a language that uses count terms. While the deduction method is, by definition, sound, it is not complete. Still on a containment corpus derived from GEOQUERY, the prover is managing to deduce many query containments automatically, without interaction. Work is underway to supply additional theorems, that when added to the stock of available theorems, will automatically solve a broader and broader class of containment problems. In the limit, it will be interesting to see how practical this approach can be made.

1. Introduction

In reviewing natural language interface efforts of the 1980s, (Copestake and Sparck Jones, 1990) made the observation that even over simple domains, systems must be based on expressive MRLs. At roughly the same time the argument was put forth on the need to be able to decide logical equivalence between arbitrary MRL expressions (Shieber, 1993). This presents something of a quandary.

Our position is to place more importance on expressive semantics than decidable query equivalence, not to say that we don't recognize the clear desirability of the later (Minock, 2014). Based on this, we must satisfy ourselves with a sound, but incomplete equivalence checkers. We justify this by appealing to the following practical problem. In our work replicating and extending PRECISE (Popescu, et. al., 2003), the lexical matching and construction of semantically tractable queries is combinatorial. Thus we tend to generate large sets of candidate queries, many of which are semantically equivalent, but syntactically distinct. It is desirable to partition these large query sets into equivalence classes, to reduce the number of queries (one from each equivalence class) that need to be reported to the user for interactive disambiguation. So even if we use a equivalence checker that has broad, but incomplete coverage, we are likely to improve the quality of interaction with the user. There are other practical examples of where a sound though incomplete query equivalence checker adds value.

As for how ambitious one wishes to be in expressivity, full first-order logic would already require us to drop completeness. But even that does not suffice; the GEOQUERY corpus, with its counts, sums and averaging questions requires reasoning beyond first-order. Furthermore SQL, with its aggregation and grouping operators is evidence that practical MRLs require reasoning over sets. Given this, we just go 'whole hog', and embrace full second-order logic.

2. Queries with counting operators

The syntax and semantics here is standard with the extra use of *don't care* existential variables (_) and the cardinality function $|X|$ returning the sizes of sets. Here we present some example questions over GEOQUERY corpus database (figure 1) paired with expressions in our higher-order query language:

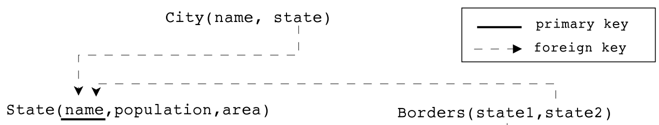

Figure 1: Part of GEOQUERY schema.

1. *"give me the cities in virginia"*:

   $\{(x)|City(x,'Virginia')\}$

   This query is a simple first-order expression which builds a set of 1-tuples for bindings of variable x.

2. *"how many cities are in montana"*:

   $\{(|X|)|(\forall x)(X(x) \leftrightarrow City(x,'Montana'))\}$

   This query returns the single tuple giving the size of the set X which is exactly the cities in Montana.

3. *"what states have more cities than ohio"*

   $$\{(x)|State(x,\_,\_) \wedge (\exists Y)(\exists Z) \\ ((\forall y)(Y(y) \leftrightarrow City(y,x) \wedge \\ (\forall z)(Z(z) \leftrightarrow City(z,'Ohio')) \wedge \\ |Y| > |Z|)\}$$

   This query, by introducing sets of cities in Ohio and sets of cities in the free variable of the query, can, via $>$ on the sizes of the sets, determine which states have more cities than Ohio.

4. *"What state has the most cites"*

   $$\{(x)|State(x,\_,\_) \wedge \\ (\exists Y)(\forall y)(Y(y) \leftrightarrow City(y,x) \wedge \\ \neg(\exists w)(\exists Z)(\forall z)(Z(z) \leftrightarrow City(z,w) \wedge |Y| < |Z|))\}$$

This query introduces a not exists over a set variable. Such quantification takes us beyond existential second order (ESO) logic.

5. *"States where the majority of cities are less than 10,000 people."*

$$\{(x)|State(x,\_,\_)\wedge$$
$$(\exists Y)(\exists Z)($$
$$(\forall y)(Y(y) \leftrightarrow (\exists p)(City(y,x,p) \wedge p < 10000))\wedge$$
$$(\forall z)(Z(z) \leftrightarrow (\exists p)(City(z,x,p) \wedge p \geq 10000))\wedge$$
$$|Y| > |Z|)\}$$

This query shows that generalized quantifiers like majority are expressible within our logic[1].

Given a database state D and a query Q of the above form, answers $Q(D)$ are computable; it is straight forward to map such queries to SQL with sub-queries computing counts. What is more problematic is deciding things like query containment and thus by extension equivalence. For example, query 3 above contains query 4. The focus of this work will be to automatically determine determine containment for as large a class of formulas as possible.

3. Using HOL LIGHT to decide query containment

Our method of testing if query Q_2 contains query Q_1 is to prepare the sentence $\Sigma \Rightarrow (Q_1 \Rightarrow Q_2)$ where Σ expresses the relevant database constraints and the *unique names assumption* for the constants in Q_1 and Q_2. If HOL LIGHT (Harrison, 2009) can prove the validity of this sentence, then containment holds[2]. If HOL LIGHT does not returns a theorem expressing the input sentence within a certain time span, then we conclude that the containment does not hold.

We also have a closely related test to determine if the size of the answer set returned by Q_2 is always greater than or equal to the size of the answer set returned by Q_1, that is $\Sigma \Rightarrow (|Q_1| \leq |Q_2|)$

4. Initial Results

To test our containment checker, we constructed a corpus of containment problems over the GEOQUERY corpus. A part of this corpus was drawn from traces of our replication and extension of PRECISE as it attempted to simplify returned query sets (See section 1). Also we are adding additional examples that more extensively exercise the higher-order capabilities of our MRL. This corpus will be available for download from the author's web site.

As it stands now, queries only requiring first order reasoning are all solved correctly. We are still trying to find counter examples, but it seems that the model elimination prover in HOL LIGHT is very much up to the task. We add simple numerical constraints to the proofs via HOL LIGHT's `ARITH_RULE` function. Determinations of answer set size constraints in count queries (e.g. query 2 above) are also being correctly solved; using the theorem `CARD_SUBSET` we can determine things such as the number of cities in the Western States is always greater than the answer to the query 2 above.

The current focus is getting the reasoner to dig into set terms. Currently we are manually constructing simple higher order proofs for individual examples to get a better insight into how to develop an automatic method.

5. Discussion

Instead of bringing in the machinery of a higher-order theorem prover (like HOL LIGHT), one might consider alternative 'approximation' methods. One method, used in PRECISE's evaluation, is to consider queries equivalent if they return the same answer over the GEOQUERY database instance. While an easy method to implement, it is unsound. At best such a method can give a sound method of determining query non-equivalence. That is, given a database state D if $Q_1(D) - Q_2(D) \neq \emptyset$ then $Q_1 \not\sqsubseteq Q_2$. One could generate a set of sample databases and hope that the probability of an unsound judgment would approach zero. Also more thought could be put into generating canonical databases for a given problem that would guarantee that witnesses in Q_1 that would not show up in Q_2. Still our plans for Σ in the future (see section 3) are to include adding things like database instance assumptions (e.g. the western states are California, Montana, Oregon, etc.). This argues argues in favor of a theorem proving based method.

Another idea is to attempt to approximate the higher-order problems in some type of first-order cover restricted to a decidable fragment and give such problems to an automatic first-order theorem prover. This was our first instinct, but it seemed to work only under very odd circumstances. There was also a lot of invented syntax that tends to muddle the problem. We have essentially abandoned this approach.

For now our project will pursue the approach outlined in this paper and encode problems in HOL LIGHT. Since humans can reason over these types of problems, a person should be able to prove such lines of reasoning and encode them in HOL LIGHT theorems to further patch the system. Such theorems need to be defined over general predicates so that the patterns of reasoning developed in one domain are useful to other domains. It will be interesting to see how practical this approach can be made.

References

A. Copestake and K. Sparck Jones. Natural language interfaces to databases. *The Natural Language Review*, 5(4):225–249, 1990.

J. Harrison. HOL Light: An Overview. In *22nd International Conference on Theorem Proving in Higher Order Logics* , Munich 2009.

M. Minock. In pursuit of decidable 'logical form'. In *Swedish Language Technology Conference (SLTC)*, Uppsala, 2014.

A. Popescu, O. Etzioni, and H. Kautz. Towards a theory of natural language interfaces to databases. In *Intelligent User Interfaces*, 2003.

S. Shieber. The problem of logical-form equivalence. *Computational Linguistics*, 19(1):179–190, 1993.

[1] Note that we extended the vocabulary to include city populations just to support this example.

[2] If the arities of the answer tuples of Q_1 and Q_2 are not equal, then we simply determine that query containment does not hold.