Answer set programming as the basis for a Homeland Security QAS

Chitta Baral
Department of Computer Sc. & Eng.
Arizona State University
Tempe, AZ 85287
chitta@asu.edu

Michael Gelfond
Department of Computer Sc.
Texas Tech University
Lubbock, TX 79409
mgelfond@cs.ttu.edu

Richard Scherl
Computer Science Dept.
Monmouth University
West Long Branch, NJ 07764
rscherl@monmouth.edu

Abstract

In this paper we discuss the applicability of the knowledge representation and reasoning language AnsProlog (Answer Set Programming) for the design and implementation of a query answering system (QAS) for homeland security. We discuss our work to date on using AnsProlog to axiomatize the travel domain. We illustrate how it can be used to represent defaults, causal relations, and other types of commonsense knowledge needed to properly answer non-trivial questions about this domain.

Introduction and Motivation

Analysts dealing with potential or actual homeland security threats have to analyze rapidly large quantities of information from a variety of sources to answer particular questions of relevance to the task at hand. Given such a scenario, we propose the use of AI technologies to:

- augment the ability of human analysts to objectively analyze large quantities of complex, oftentimes ambiguous or contradictory data while simultaneously reducing the impact of their personal biases.
- to facilitate information/knowledge sharing and semantic understanding while avoiding cognitive overload.

The 9/11 Commission Report(National Commission on Terrorist Attacks Upon the United States 2004) details the extensive travels of the terrorists when preparing for their attacks. Meetings were arranged in locations such as Hamburg, Madrid, and Phoenix. We can retrospectively imagine an analyst querying this data to determine who has met with whom at which times. And when the information was added that that Khalid Sheikh Mohammed's nickname was Mukhtar (p. 277), the two individuals could then be identified revealing immediately new connections to the analyst.

We envision a query answering system (QAS) consisting of a *search engine* which searches diverse sources for information relevant to the given query, Q; a *natural language processing module* (NLPM), which translates this information (including the query) into a theory, F, of some knowledge representation language $\mathcal{L}$; a *general knowledge base*, KB, containing common-sense and expert knowledge about various domains; and an *inference engine* which takes F and KB as an input and returns an answer to Q. Even though the choice of the KR language $\mathcal{L}$ is irrelevant for the first component of the system it plays an increasingly important role in the design of its other components. In this paper we hypothesize that AnsProlog - a language of logic programs under the answer set semantics (Gelfond & Lifschitz 1988; 1991) - is a good candidate for the KR language of QAS. This is especially true if answering a query Q requires sophisticated kinds of reasoning including default, causal, and counterfactual reasoning, reasoning about narratives, etc.

The list of attractive properties of AnsProlog include its simplicity and expressive power, ability to reason with incomplete information, existence of a well developed mathematical theory and programming methodology (Baral 2003), and the availability of rather efficient reasoning systems such as SMODELS(Niemela & Simons 1997). AnsProlog allows its users to encode defaults, causal relations, inheritance hierarchies, and other types of knowledge not readily available in other KR languages. In addition it supports construction of elaboration tolerant knowledge bases, i.e., ability to accommodate new knowledge without doing large scale surgery. The main drawback of the language is the inability of its current inference engines to effectively deal with numbers and numerical computations.

Representing general knowledge

As an example of our approach, we have already built a commonsense theory of travel in AnsProlog. A basic object in the theory is a trip. Trips have origins, destinations, and intermediate stops. A trip can have many participants and may make use of vehicles of different types. The preconditions of a participant jointing a particular trip include the possession of appropriate documents such as tickets, visas, and passports.

Central to the theory are the causal laws that specify the effects of traveling events on the locations of persons and objects. We are mainly interested in locations of people and in various traveling events which change these locations. Construction of the theory is based on the theory of dynamic systems which views the world as a transition diagram whose states are labeled by fluents (propositions whose values depend on time) and arcs are labeled by actions. For instance, states of the diagram, D, can contain locations of different

people; a transition $\langle \sigma_0, \{a_1, a_2\}, \sigma_1 \rangle \in D$ iff σ_1 is a possible state of the domain after the concurrent execution of actions a_1 and a_2 in σ_0. There is a well developed methodology of representing dynamic domain in AnsProlog (Baral & Gelfond 2000; Turner 1997) which has been used in a somewhat simplified form.n

This theory also interacts with other modules. One module contains geographical information about cities, countries, regions of the world and the relationship between them. This module provides information about which cities are in which countries and which countries are in which regions of the world. We assume that the theory is complete so that negative information can be concluded via the closed world assumption. Additionally, a theory of calendrical and temporal information is also needed. This module contains axioms about days, months, years, seasons and their relationships.

Using these theories we can translate various scenarios into our AnsProlog notation and answer queries. Consider the following scenario and query:

> John took the plane from Paris to Baghdad. On the way the plane stopped in Rome. Where is John?

The scenario is translated as follows:

```
h(at(john, paris),0).
o(go_on(john,f(paris,baghdad)),0).
mode_of_transp(f(paris,baghdad),air).
o(stop(f(paris,baghdad),rome),2).
```

Here a particular trip from paris to baghdad is named `f(paris,baghdad)`. The predicate `h` is used to indicate the truth of a particular fluent at a particular time point and `o` is used to indicate the occurrence of an action.

The query is:

```
h(at(john,baghdad),n).
```

Here `n` is used to represent a later point in time. The translation of the scenario and query into AnsProlog correctly yields the answer "yes" when combined with our commonsense theory of travel, since trips (along with their participants) generally by default continue to their destination as indicated by the following default.

```
o(stop(F,D),T) :-  h(at(F,en_route),T),
                   dest(F,D),
                   not -o(stop(F,D),T).
```

The default rule essentially states that a trip will stop at a particular destination at a particular point in time if there is no reason to conclude (default negation – `not`) that the action does not occur (logical negation `-`). Other axioms specify that if a stop action occurs, all participants in the trip are at the location at the next point in time.

On the other hand, consider a slightly different scenario and query:

> John took the plane from Paris to Baghdad. On the way, the plane stopped in Rome, where John was arrested. Is John in Baghdad?

Now we obtain the answer "No" since the default was overridden by the "arrest" action. An arrest axiom is axiomatized as removing the object of the arrest as a participant in the trip.

The current system can also answer queries involving dates and typical durations of trips. Additionally, we have developed a planning module to determine if an actor can (given a particular set of facts) be in a particular location by a particular date. Furthermore, to a limited extent the theory allows reasoning about plans or intentions of actors.

Current Work

Our current work is focused on a number of issues including:

- The development of reasoning systems that combine AnsProlog modules with Prolog modules to overcome the weakness of AnsProlog inference engines in dealing with numbers and numerical calculations.
- Extending the work reported above to handle more complex queries such as counterfactual reasoning and plan/intention recognition.
- Automating the translation of English scenarios and questions into AnsProlog.
- Adding further theories to handle reasoning about weapons of mass destruction and monetary transactions.

Summary

In conclusion, we feel that the features of AnsProlog are well suited to form the foundations for an inference engine supporting a QAS. Our future work will develop the support tools and implementation needed to demonstrate this hypothesis.

Acknowledgments

This work has been supported by ARDA under award number 2004-H900700-000.

References

Baral, C., and Gelfond, M. 2000. Reasoning agents in dynamic domains. In Minker, J., ed., *Logic Based AI*. Kluwer.

Baral, C. 2003. *Knowledge representation, reasoning and declarative problem solving*. Cambridge University Press.

Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. In Kowalski, R., and Bowen, K., eds., *Logic Programming: Proc. of the Fifth Int'l Conf. and Symp.*, 1070–1080. MIT Press.

Gelfond, M., and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9:365–387.

National Commission on Terrorist Attacks Upon the United States. 2004. *The 9/11 Commission Report*. W.W. Norton & Company.

Niemela, I., and Simons, P. 1997. Smodels – an implementation of the stable model and well-founded semantics for normal logic programs. In *Proc. 4th international conference on Logic programming and non-monotonic reasoning*, 420–429.

Turner, H. 1997. Representing actions in logic programs and default theories. *Journal of Logic Programming* 31(1-3):245–298.