Answering Science Questions: Deduction with Answer Extraction and Procedural Attachment

Richard Waldinger

Artificial Intelligence Center
SRI International
Menlo Park, CA 94025
waldinger@ai.sri.com

Jeff Shrager

CommerceNet and Symbolic Systems Program
Stanford University
Stanford, CA 94305
jshrager@stanford.edu

Abstract

An approach to question answering through automated deduction is advocated. Answers to questions are extracted from proofs of associated conjectures over an axiomatic theory of the subject domain. External knowledge resources, including data and software, are consulted through a mechanism known as procedural attachment. A researcher ignorant of the subject domain theory or its logical language can formulate questions via a query elicitation facility. A similar device allows an expert to extend the theory. An English explanation for each answer, and a justification for its correctness, is constructed automatically from the proof by which it was extracted. A deductive approach has been applied in planetary astronomy, geography, intelligence analysis, and, most recently, molecular biology and medical research applications. It is argued that the constructs in the Semantic Web languages, including OWL with SWRL, are insufficiently expressive for this kind of application.

The Short Story

We advocate a combination of several deductive technologies as an approach to semantic knowledge integration in the sciences. The particular technologies we deploy include

- Axiomatic subject-domain theory: Knowledge of the subject domain is expressed by an axiomatic theory, in a logical language.
- Answer extraction: A query (or other task) is paraphrased as a conjecture in the language of the theory and proved using an automatic theorem prover; an answer to the query is extracted from the proof.
- Procedural attachment: Selected symbols of the theory are linked to local or Web-based external resources, including data, software, and other knowledge. When such a symbol participates in the proof search, the external resource is consulted. Information provided by the resource can then be incorporated into the proof search.
- Query elicitation: A researcher, who may be ignorant of formal logic or the subject domain theory, is guided in formulating a query that is within the domain of competence of the theory. The researcher makes choices between ever-expanding alternative query fragments permissible according to the sort or type structure of the theory. The researcher sees the English query; the theorem prover sees the corresponding logical form.
- Knowledge elicitation: The mechanism behind query elicitation also allows a subject domain expert who is ignorant of logic or the existing theory to extend the subject domain theory, incorporating new knowledge by defining it in terms of existing knowledge.
- Unreliable resources: The information used in the proof is accompanied by an estimate of its reliability; more reliable information is to be preferred, and the answer is accompanied by its own reliability estimate.
- Proof-based explanation and justification: A coherent explanation of and justification for the answer is constructed automatically from text supplied with the axioms used in the proof. This also conveys the provenance of (and a reference list for) the answer.

The approach is domain independent and has been applied to a number of domains, by ourselves and others. In this paper we will illustrate it with an example from molecular biology, for which a proof-of-concept implementation, called BioDeducta, has been developed. This example and others suggest that current Semantic Web technology may not meet the needs of scientific question answering.

Purpose

We are interested in problems whose solution must be deduced from information provided by many disparate resources, which can include data, software, and knowledge bases. These resources, which may have been developed by diverse distributed people or institutions, are not intended to work together and may have adopted inconsistent representations and conventions. The intended user is a researcher who is not primarily a computer scientist, and who furthermore may be ignorant of the resources appropriate to solve the problem at hand. Not all the resources are equally trusted; more reliable resources are to be preferred, and the answer itself is to be accompanied by a reliability estimate. The answer is also to be provided with an explanation, which is a justification of its correctness, and its provenance, which is an indication of its sources.

Approach

Our approach is based on deductive inference. We require the development of an axiomatic description of the subject knowledge, called the subject domain theory. This theory contains sentences in the language of logic that describe the concepts in the query, the capabilities of the available external resources, and the background knowledge necessary to link them together.

The query or other task is phrased as a conjecture a theorem to be proved in the logical language of the subject domain theory, and submitted to an automatic theorem prover. Typically, entities to be found are represented by existentially quantified variables; the existence of entities that satisfy the conditions of the query is conjectured. To prove the conjecture, the theorem prover must replace these variables with concrete computable terms that satisfy the conditions. The theorem prover keeps track of these replacements and uses them to construct an answer, which is extracted from the proof. Many proofs are possible, and each may yield a different answer.

Certain of the relation and function symbols in the subject domain theory are linked to external resources. When any of these symbols appears in the proof search, the corresponding resource may be consulted. Information provided by the resource may then be entered into the proof search, just as if it had been represented axiomatically. Typically, the ultimate answer is composed of information provided by a collection of external resources. As a consequence, the theory becomes not so much a repository of knowledge as an index to where knowledge can be found and how it can be used.

The information provided by one external resource is likely to be in a different form from that required by another. If a resource is available that can translate from one form into the other, it may be invoked by the same inference-plus-procedural-attachment mechanism that invokes any resource. This provides a principled approach to achieving interoperability among disparate resources.

In general, theorem proving is an undecideable combinatorial search problem; in the absence of strategic knowledge, a theorem prover is likely to explore a very large search space before finding a solution to a difficult problem. Within a particular subject-domain theory, however, we can formulate search strategies that enable a theorem prover to exhibit high performance. In particular, we may apply an ordering to the symbols of the theory, where a lower-ranked symbol is to be preferred to a higher-ranked one. This then restricts the theorem prover to perform operations that replace the higher-ranked symbols with the lower-ranked ones. Similarly, we can apply numerical weights to the symbols that allow us to compute a weight for every deduced formula. The theorem prover will give first priority to the lighter formulas and defer work on the heavier ones. By introducing appropriate orders and weights we can give the theorem prover a sense of direction and achieve dramatic orders-of-magnitude improvements in proof search efficiency. Later we argue that this is a more appropriate approach than restricting the language in which we express axioms and queries.

Not all knowledge is equally reliable. It is expected that uncertain knowledge may be accompanied by a confidence estimate. When a new conclusion is deduced, its confidence estimate will be computed based on the reliability of the knowledge on which it is based. (For instance, if the confidences are real numbers between 0 and 1, they may be multiplied, but there is no reason that the confidences have to be numerical, or even one-dimensional.) Once one answer has been discovered, we can continue searching for other, perhaps more reliable, answers.

It is not realistic to expect that researchers will formulate their queries directly in the language of logic. On the other hand, if we allow arbitrary English queries, it is difficult to constrain the researcher to stay within the capabilities of the subject domain theory and to use the theory's vocabulary in the expected way. Rather, we provide a query-elicitation mechanism to guide the researcher in formulating queries that the subject domain theory can understand. The subject domain theory is sorted; entities are classified according to their types or sorts, and relation and function symbols are declared to restrict the sorts of their arguments and values. If the researcher begins the query-formulation process by mentioning an entity, the mechanism offers a choice among the function and relation symbols that could apply to that entity. While internally these choices are represented in logic, the choices

presented to the researcher are expressed in pre-stored English. Once the researcher has made a choice, the mechanism offers ways of extending the choice, either by filling in arguments or by wrapping other function or relation symbols, or logical connectives, around the query fragment already formulated. At any stage, the researcher may request sample answers to the query so far, which provide feedback to help refine the query.

The mechanism we propose for query elicitation may also be used to acquire knowledge from a subject-domain expert who is unfamiliar with the language of logic or the content of the existing subject-domain theory. (Indeed, formulating a complex query often requires one to impart the knowledge that comprises the hypotheses or premises of the query.) In acquiring knowledge, we must allow the expert to define and use new vocabulary. Knowledge acquisition is particularly useful in enabling the provider of an external resource to describe that resource with logical axioms.

In logical form, the proof is not readable except, with some difficulty, by specialists. It contains enough information, however, to allow us to construct a coherent English explanation of the answer, a justification of its correctness, and an indication of the sources on which it is based. A simple way to do this is to provide, with each entity and axiom in the subject domain theory, an English paraphrase of its meaning. Axioms that are regarded as obvious may be given no English paraphrase at all. The explanation and justification is constructed by concatenating the English paraphrases corresponding to the axioms of the proof, in the order in which they appear in the final proof. Variables in the axioms are replaced by the English paraphrase of the terms with which they are replaced. We have found it generally unnecessary to attempt to paraphrase the logical inferences performed during the proof; usually the axioms alone are enough explanation. Since the proof contains an indication of the external resources that have been invoked by the procedural-attachment mechanism, it forms the basis for constructing a reference list, which provides the provenance of the answer.

We have applied this domain-independent approach in planetary astronomy (Stickel et al. 1994), geography (Waldinger et al. 2002), intelligence analysis (Waldinger et al. 2004), and molecular biology (Waldinger and Shrager 2006; Shrager et al. 2007). We are currently also looking at the querying of medical research data. Note of these systems yet include query elicitation, proof-based explanation, or confidence estimates. Axioms were encoded in logic by hand; in each case, SNARK was the deductive inference engine.

Of all of these, the system Amphion, developed with our assistance by NASA to allow planetary astronomers to query data from space missions, has so far been the most practical it is used to plan photography and analyze data from the Cassini mission to Saturn. See http://ic.arc.nasa.gov/ic/projects/amphion/

We shall consider a case study from the molecular biology domain.

Example

Let us consider part of an example performed by our proof-of-concept prototype BioDeducta, which implements the approach. This system depends on a subject domain theory for parts of molecular biology, expressed in a sorted first-order logic, and uses SNARK (Stickel, Waldinger, and Chaudhri 2000) as its theorem prover. Procedural attachments access data and software residing in the BioBike (Massar et al. 2005) environment for biological computing.

The problem originally formulated by biologist J. Elhai concerns cyanobacteria, a class of very common waterborne bacteria that can perform photosynthesis. Some of these bacteria (e.g., Prochlorococcus sp. strain Med4, here called promed4) live near the surface of the sea and are adapted to a high light level. Others (e.g., Prochlorococcus sp. strain MIT9313, here called mit9313) live deeper below the surface and are adapted to a somewhat lower light level. Biologists are actively interested in determining which genes are involved in this adaptation to differing light levels. We may refine this question by asking to find, in promed4, a gene that is light sensitive in microarray experiments and that has no ortholog in mit9313. Since mit9313 is not light adapted, such a gene is very likely to be involved in the adaptation to light.

In a microarray experiment, we submit all the genes of an organism to a stimulus (in this case light) and see which of them exhibit a significant change in RNA production (either increased or decreased). Two genes in different organisms are orthologs if they are descended from the same gene in a common ancestor. Typically, orthologs share a common function; thus, if a gene is light sensitive in one organism, its ortholog will be light sensitive in another.
The logical form of our query is

```
gene-in-organism(?gene1, promed4)
 & function(light, ?gene1, promed4)
 & (forall(gene2)
   (not
    [ortholog(?gene1,gene2) &
     gene-in-organism(gene2,mit9313)]))
```

The query is regarded as a conjecture a theorem to be proved and submitted to the theorem prover. The variable `?gene1` in this query has tacit existential quantification, and, by a notational convention, ranges over genes. This means we want to find a gene `?gene1` of `promed4` that

is light sensitive and that has no ortholog `gene2` in `mit9313`.

The conjecture is rewritten by the application of axioms from the subject domain theory. For example, the axiom

```
function(?stimulus, ?gene, ?organism)
     <=
experiment(?stimulus,
           ?gene, ?organism, high)
```

states that the function of a gene in an organism may be determined by examining its behavior in a microarray experiment. Because of a duality between queries and assertions, variables in assertions have universal, not existential, quantification the assertion holds for all stimuli, genes and organisms.

The axiom

```
  ortholog(?gene1, ?gene2)
& gene-in-organism(?gene2, ?organism2)
     <=>
gene-has-ortholog-in-
     organism(?gene1,?gene2,?organism2)
```

allows us to rephrase the conjunction of two conditions in the query by a single condition. The value of this rewriting is that the symbol `gene-has-ortholog-in-organism` has a procedural attachment that allows us to search for orthologs of a given gene in a given organism.

We will not go through the solution discovered by BioDeducta in detail; this is done in other papers (e.g., (Waldinger and Shrager 2006; Shrager et al. 2007). Certain of the conditions are solved by procedural attachment to an external resource the National Center for Biotechnology Information and CyanoBase maintain files of genes in various cyanobacteria and are accessible through BioBike. BioBike also allows access to tables of orthologs, to microarray data, and to tools for analyzing this data, which enables SNARK to produce a list of genes that satisfy the conditions. In a test run, SNARK required about twenty seconds to find one gene that satisfied the conditions, and another twenty seconds to find that there were no others. The use of numerical weights was crucial in the efficient search for this solution. Lower weights were given to symbols whose procedural attachments generally returned fewer answers, resulting in a more constrained search. Without appropriate strategic controls, the same query requires hours.

In the next section, we discuss some issues that are highlighted by this case study.

What does a Question-Answering Logic Need?

Much work in ontology development for scientific applications has proceeded without considering how the ontologies are ultimately to be used. The light-adaptation example illustrates some features required by a query language and logic for deductive question answering.

The query contains both universal and existential quantification. The variable `?gene1` is essentially existential, since it represents an entity whose existence must be established, while `gene2` is universal the condition must hold for every gene `gene2`.

We require true negation, not negation as failure. We want to establish that each gene `gene2` is not an ortholog, not that we have simply failed to establish that it is.

We employ a closed-world assumption for the ortholog data. When we fail to find an ortholog for `?gene1` in the table, we conclude that none exist, because our external resources maintain a complete list of orthologs.

We mention these features of the reasoning because they are awkward, and perhaps impossible, to reproduce within the framework of OWL (McGuinness and van Harmelen 2004) and SWRL (Horrocks et al. 2004), which are leading candidates for a Semantic Web inference framework. Other features that are essential, but not highlighted by this discussion, include the presence of logical disjunction, function symbols, and equality reasoning. (Actually, in resolution theorem provers, including SNARK, the universal quantifier in the query is paraphrased using a Skolem function symbol, but function symbols do not exist in OWL/SWRL either.) While it is perhaps possible to reproduce this kind of query by other means, it would require a lot of circumlocution. First-order logic contains just enough strength to allow a fairly direct representation of the question, while still admitting an efficient solution.

This example and others that we have worked with suggest that the full expressive power of at least first-order logic is necessary for the direct expression of scientific knowledge and queries. Efficient processing of queries might better be achieved by the development of subject-specific control strategies than by the unnatural restriction of knowledge-representation languages.

Acknowledgements

We thank J. P. Massar and Mike Travers for their work on the implementation of the BioBike system; Mark Stickel for the development of the SNARK theorem-proving system and assistance with its use; Carolyn Talcott, Mark-Oliver Stehr, and Steve Racunas for work on the axiomatic theory of biological pathways; Amar Das, Martin O'Connor, Noah Zimmerman, and Genaro Hernandez for

advice on the axiomatic representation of medical research data.

This research was supported in part by the National Science Foundation, under the Science and Engineering Information Integration and Informatics (SEIII) Program (NSF IIS-0513857), and by a grant from the NASA Advanced Information Systems Research Program through cooperative agreement NO6- 4803 between NASA/Ames (Andrew Pohorille P.I.) and The Institute for the Study of Learning and Expertise.

References

Horrocks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosof, B., and Dean, M. 2004. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. Member Submission, W3C. http://www.w3.org/Submission/SWRL/

McGuinness, D. and van Harmelen, F., eds. 2004. OWL Web Ontology Language Overview. Technical report. W3C. http://www.w3.org/TR/owl-features/

Massar, J. P., Travers, M., Elhai, J., and Shrager, J. 2005. BioLingua: A Programmable Knowledge Environment for Biologists. BioInformatics, 21(2): 199-207.

Shrager, J., Waldinger, R., Stickel, M., and Massar, J. 2007 Deductive Biocomputing. PLoS ONE, vol. 2, no. 4, pp. e339.

Stickel, M., Waldinger, R., Lowry, M., Pressburger, T., and Underwood, I. 1994. Deductive Composition of Astronomical Software from Subroutine Libraries, in 12th Conference on Automated Deduction, Nancy, France. In Automated Deduction, A. Bundy, ed., Springer-Verlag Lecture Notes in Computer Science, Vol. 814.

Stickel, M., Waldinger, R., and Chaudhri, V. 2000. A Guide to SNARK. SRI International, Artificial Intelligence Center http://www.ai.sri.com/snark/tutorial/tutorial.html

Waldinger, R., Reddy, M., Culy, C., Hobbs, J., and Dungan, J. 2002. Deductive Response to Geographic Queries, in GIScience 2002, Boulder, CO,.

Waldinger, R., Appelt, D., Fry, J., Israel, D., Jarvis, P., Martin, D., Riehemann, S., Stickel, M., Tyson, M., Hobbs, J., and Dungan, J. 2004. Deductive Question Answering from Multiple Resources. In New Directions in Question Answering, M. Maybury, ed., AAAI.

Waldinger, R. and Shrager, J. 2006. Deductive Discovery and Composition of Resources. Reasoning on the Web: Workshop at the 15th International World Wide Web Conference. Edinburgh, Scotland. http://www.aifb.uni-karlsruhe.de/WBS/phi/RoW06/procs/waldinger.pdf