Expanding the Notion of Answer in Rule-Based Systems

Topic Area: Representational Formalisms

Keywords: Question Answering

Debra T. Burhans and Stuart C. Shapiro
Department of Computer Science and Engineering
and Center for Cognitive Science
State University of New York at Buffalo
226 Bell Hall
Buffalo, NY 14206-2000
burhans@cse.buffalo.edu, shapiro@cse.buffalo.edu

November 1, 1999

Expanding the Notion of Answer in Rule-Based Systems

1 Introduction

The traditional notion of a question in AI is an open sentence Q(x1..xn), and the traditional notion of an answer is a set of a1...an such that Q(a1...an). More recently, this notion of answer has been extended to generic answers of the form

```
All(x1...xn)[G(x1...xn) => Q(x1...xn)].
```

We further extend the notion of answer to include hypothetical/generic answers of the form

```
(Exists(w1...wn1)H(w1...wn1))
=> All(x1...xn2, y1...yn3)[G(x1...xn2, y1...yn3)
                           => All(z1...zn4)Q(x1...xn2, z1...zn4)].
```

We formally show how every clause generated during the course of a refutation resolution procedure may be analyzed as a hypothetical/generic answer, as long as it descends from the query clause. Informally, in the above schema: Q, the specific part of the answer, represents literals that were part of the query; G, the generic part, represents literals that share variables with the Q literals, or that share variables with other G literals; and H, the hypothetical part, represents literals whose variables don't occur in either G or Q. Each part may also contain constants that were or weren't part of the query.

2 Background

The role of resolution theorem proving in question answering was established by Cordell Green [6, 5] with the introduction of the answer literal. This literal contains the variables from the query and is added to the clause(s) corresponding to the negation of the query. If the resolution refutation procedure produces the empty clause, the variable bindings found along the way are captured in the answer literal.

The goal of resolution refutation in this case is the production of the empty clause: absent that, no answers will be produced. This is the approach taken by Prolog and resolution theorem provers. The type of answer produced using this approach is termed extensional, or specific, and the form of such an answer can be characterized as a set of a1...an such that Q(a1...an).

Cholvy and Demolombe [2, 3] expanded upon Green's work by looking at resolution in a rule base with no ground terms. In this situation, the empty clause is never produced, yet answers in the form of rules rather than facts are discovered. Such answers are termed intensional or generic. The general form of these answers is:

```
All(x1...xn)[G(x1...xn) => Q(x1...xn)].
```

Motro [7, 8, 9] has examined the problem of intensional answering in the context of databases. His databases contained both rules and facts and his answers could contain elements of both extensional and intensional answers, in which case he used the term *mixed*.

While generic answers have been described in the AI and database literature, applications that use resolution refutation are focused on the search for the empty clause, and hence, carry on the tradition of specific question answering. In addition, specific answering is the paradigm found in introductory AI textbooks [12, 11, 10, 4].

3 Recognizing Specific and Generic Answers

The criterion for recognizing specific answers is clear: when the empty clause is produced during the course of a resolution refutation proof, the variable bindings in effect at that time comprise a specific answer. If the option to continue a proof beyond the point where the empty clause is derived exists, the next resolution step that derives the empty clause produces another specific answer, and so on.

The criteria for recognizing a generic answer arise out of a concern for the relevance of an answer. If a clause contains an answer literal and one or more additional literals and the variables in the answer literal overlap completely with the variables in the other literals, it is clear that all the literals in the clause are relevant to the query. A clause with this form constitutes a generic answer. The form of the generic answer will be the conjunction of

the negation of the non-answer literals followed by an implication symbol, followed by the answer literal. Thus, the characterization of generic answers as rules. The variables in a generic answer, shared by the answer and non-answer literals, are assumed to be universally quantified.

Generic answers are generated along the way to finding specific answers. In case there are no specific answers, there may still be generics. The question then becomes, what is generated along the way to finding generic answers, and do these resolvants represent some other type of answer.

4 Hypothetical Answers

Consider the clauses that represent neither generic nor specific answers. Such clauses, provided they descend from the original query, contain at least one answer literal along with a non-empty set of non-answer literals that do not share variables with the answer literals. Such clauses are termed hypothetical answers [ref to our fall symp paper]. The interpretation by some other researchers of these clauses is that they are uninteresting because they will be subsumed by generic or specific answers. That is, additional literals containing variables not in the answer literals have been regarded as either not relevant, or not interesting because they will later be subsumed. A clause containing such additional literals can be represented as follows: the negation of the conjunction of the "extra" literals can be taken as the left hand side of an implication, where the right hand side of the implication is either a generic or a specific answer.

The following example (Example 1.) is presented in order to illustrate hypothetical answers, and how they relate to specific and generic answers. Consider this simple rule base:

```
all calicos are cats
fluffy is a calico
rover is a horse or rover is a dog
calicos like dogs
calicos do not like horses
```

and the question, *is there something that fluffy likes.*

The following comprise the answers in the order produced. Each is followed by a brief explanation.

```
((CALICO FLUFFY)) =>
  (All (?x0)((DOG ?x0))) =>
     ( (LIKES FLUFFY ?x0))
```

A gloss of this is *if fluffy is a calico, fluffy likes dogs.* The hypothetical portion of the answer contains only constants, and the generic and specific portions of the answer share a variable. The constant fluffy is also found in the specific part of the answer. It is variable sharing that structures the answer, regardless of the presence of constants.

The second answer produced is:

```
  (All (?x0)((DOG ?x0))) =>
     ( (LIKES FLUFFY ?x0))
```

A gloss of this is *fluffy likes dogs.* This answer shows what happens when the hypothetical portion of the previous answer has been "discharged", that is, the hypothetical portion of the answer has been eliminated by the process of resolution.

The third answer is:

```
((CALICO FLUFFY) &  (~ (HORSE ROVER))) =>
     ( (LIKES FLUFFY ROVER))
```

A gloss for this answer is *if fluffy is a calico and rover is not a horse, then fluffy likes rover.* There is no generic part to this answer, and there are no variables. The specific part of the answer is simply the answer literals, and the rest of the literals unrelated by variable sharing comprise the hypothetical portion.

A fourth answer is:

```
( (~ (HORSE ROVER))) =>
     ( (LIKES FLUFFY ROVER))
```

A gloss for this is *if rover is a horse then fluffy likes rover.* This is simply the third with the hypothetical discharged, as described above: it is known that fluffy is a calico.

Finally, a seemingly unusual answer is produced:

```
(Exists (?x0)((CALICO ?x0) & (LIKES ?x0 ROVER))) =>
     ( (LIKES FLUFFY ROVER))
```

It was in fact this example that led directly to our general formulation of the form of an answer. In an older system, this was rejected as not relevant. A gloss is *if there is a calico that likes rover, then fluffy likes rover.* This is a perfectly reasonable and correct answer, though it falls far from what we commonly consider as an answer.

5 A General Characterization of an Answer

Based on the foregoing discussion, it is clear that the literals appearing in a clause can be partitioned into three groups. First, the answer literals, second, the non-answer literals involved in generic answers, and third, the non-answer literals that have no variables in common with either the answer literals or the generic literals. While the first and third of these groups can be easily identified, the second must be carefully characterized. The collection of literals "involved in generic answers" can be more precisely defined as follows. Included in this set are all literals in the variable closure of the answer literals, where this closure is defined as: Literals that share any variable with some answer literal are in the set, and in addition, literals that share any variable with any literal in the set are included.

We apply the terms specific, generic, and hypothetical to the three groups of literals characterized above, where the specific portion of an answer corresponds in most respects to what has previously been termed specific, and similarly for the generic portion of an answers. We expand the generic portion of the answer by including the closure of the variables in the answer rather than simply the variables. The hypothetical portion of the answer has been described in [1]. The following form shows how the different parts of the answer fit together:

```
(Exists(w1...wn1)H(w1...wn1))
  => All(x1...xn2, y1...yn3)[G(x1...xn2, y1...yn3)
                             => All(z1...zn4)Q(x1...xn2, z1...zn4)].
```

In addition, each part may also contain constants, where these constants may or may not have been part of the original query.

It is clear that specific, generic and mixed answers as previously defined fit into this framework. In the case of specific answers, only the rightmost term is present (the specific portion), and it contains only constants, no variables.

Generic answers comprise the middle and rightmost terms (the generic and specific portions), and contain only variables that appear both in the generic and specific parts. Mixed answers are simply generics in which constants appear in either the generic or specific part. Hypothetical answers are those containing a hypothetical component. The only required component of an answer is the specific portion, reflecting the fact that the answer literals are central to this process of question answering. It is the specific portion that connects the products of resolution with the original query.

6 All Resolvants that Descend from the Original Query are Answers

The procedure for using resolution refutation as a question answering mechanism involves adding an answer literal to the negation of the original query (which might yield more than one clause, meaning each would contain an answer literal). Using the set of support strategy, and setting the initial set of support to the negation of the query, resolution begins by looking for a clause to resolve with the negation of the query. The only way for a clause to be added to the set of support is for it to be the resolvant of some clause from the set of support with some other clause. The answer literals are completely ignored by the resolution process and are merely "carried along" in clauses with the other literals. In this variant of resolution refutation, the empty clause is, in fact, a clause with only answer literals.

More formally:

The set of support strategy guarantees that all clauses generated during the course of a resolution refutation proof descend from the set of support. Namely, each new resolvant has one parent that is a supported clause.

The set of support initially contains the negation of the query with the added answer literal(s).

The first resolvant produced will have as a parent the negated query clause, which contains an answer literal. Since the answer literal is ignored by resolution, the resolvant will contain the intact answer literal inherited from the parent, with updated variable names as necessary to resolve the clauses.

This resolvant is placed in the set of support, which clearly still contains

only clauses containing an answer literal.

Each step of resolution repeats this process, selecting one clause from the set of support, and another from the set of clauses in the rule base.

Therefore it is impossible to produce a resolvant that does not contain an answer literal.

Therefore producing resolvants in this manner means that every resolvant can be considered an answer. The presence of the answer literal ensures that every resolvant is descended from the original query, which was initially the only clause(s) containing an answer literal. In addition, it is clear that all such resolvants are relevant to the query.

6.1 A New Form for the Answer Literal

The way in which we construct the answer literal differs from past approaches. Consider a query `P(x1, ..., xn)`. Rewrite the query as the antecedent of the answer literal as follows:

```
All(x1, ..., xn)[P(x1, ..., xn) => ANSWER(P(x1, ..., xn)]
```

In this form, the query is negated, and this can now be converted to clause form and added to the rule base.

7 Relationships Between Hypotheticals, Generics, and Specifics

While in one sense it is reasonable to view as hypotheticals those answers found along the way to generics or specifics, and generics as those answers found along the way to specifics, the relationship between hypotheticals and the other types of answers is fundamentally different than that between generics and specifics.

It is better to know whether something is or is not the case rather than to be left with uncertainty. Therefore, there is a sense in which the task of settling the question of whether the hypothetical portion is in fact the case is of critical importance. This will be termed "discharging the hypothetical".

On the other hand, there is no analogous "discharging the generic" process, precisely because a generic is a desirable answer and represents information contained in the knowledge base. The only reason pursue a more

specific answer, which in a sense "discharges" the generic, is when the goal is to obtain a specific rather than a generic answer. There is no existential presupposition associated with generics, so answers such as "all floobles squonk" carries no entailment about the existence of floobles. There are clearly cases when a specific answer is desired, and in such cases generics should be discharged as quickly as possible.

A generic captures what a set of specific answers have in common, and does so often in a succinct and clear manner. The same simply can not be said for a hypothetical.

8 The "Discharging Hypotheticals" Search Strategy

The example given above expressed, in the hypothetical portion of the answer, the question of whether or not fluffy was a calico. Once a hypothetical has been discharged, the information should not simply be forgotten, only to be retrieved from scratch at a later time. A search strategy of "discharging hypotheticals" is proposed that will serve two purposes. First, information relevant to hypotheticals that have already been discharged will be cached for easy later retrieval. Note that this is analogous to the indexing of predicates performed by Prolog. Second, clauses in the set of support will be ordered so that those with top priority for resolution will be the hypotheticals.

The set of support is an ordered list of clauses, where all hypotheticals come at the beginning of the list, and within the hypothetical and non-hypothetical portions of the list any ordering desired can be implemented, such as shortest clause first, most recently generated clause first, etc.

An outline of the search strategy is as follows:

```
choose clause to resolve from the front of the supported
   clauses list
if clause is a hypothetical
  check the list comprising the cached information
    of hypotheticals already resolved to see if the
    hypothetical portion of the clause can be immediately
    discharged
  if hypothetical can be discharged, do so, and place
```

```
    the resolvant back in the list of supported clauses
    in the appropriate position
  else
    try to resolve just the hypothetical portion of the
    clause with other clauses in the rule base

    in case of success place the resolvant back in
      the list of suppported clauses as described above for
      use in subsequent resolution
else
  proceed with resolution as usual, placing resolvant(s)
  in the appropriate positions in the list of supported clauses
```

The motivation for developing this strategy arises from the need to provide good answers to questions, but the effect may be beneficial for other problems that use resolution refutation as a reasoning strategy.

When the hypothetical concerns attributing properties to some object, the act of creating a list of properties associated with that objects starts to acquire the flavor of description logic, only the association of properties with objects is driven by particular queries and is not fundamental to the data structure.

For the small problems on which we have tested this strategy there has not been a notable difference in performance of the system. However, we are planning more extensive experiments where this strategy can be compared with other standard search strategies.

The list of information used to discharge the hypotheticals may itself be interesting. That is, knowing that the fact that fluffy was a calico was critically important in answering a particular question might help in the formation of future queries, including possibly reformulating questions so that not as many hypotheticals are generated.

9 Information from Hypotheticals

A specific answer is a witness that proves the truth of an existential hypothesis: we have considered questions having this form. If you ask about dogs, and Fido is a known dog, Fido will be involved in your answer. A generic

answer is a rule, capturing generalities about classes or groups of objects. What, then, is the purpose or utility of a hypothetical answer?

The hallmark of the hypothetical answer is the way in which information belonging to the hypothetical portion is identified. That is, variables not shared with the answer literal, nor in common with those in the generic portion of the answer. If information to discharge the hypothetical is not available in the rule base, this indicates an information deficiency, or underspecification of the question. In some cases the hypothetical can serve as a useful tool to a rule base designer. For example, if you are trying to prove a theorem and the answer you get back is a hypothetical rather than the expected "yes" or "no", it may be a sign that an important piece of information has been left out. In other cases you may purposely want to ask an underspecified question. For example, if you query a rule base about restaurants, and don't specify which sort of cuisine, hypothetical answers could include information such as if you like French food, go to Cafi Boeuf.

10 What is "the answer"?

It would seem annoying and uninteresting to return as answers clauses that have been subsumed by other clauses. On the other hand, specifics subsume generics, and both might be interesting and useful answers. The notion of best answer is certainly relative to the person asking the question. If a preference for a particular type of answer, or a desire to avoid certain types of answers is expressed, this can be built into a system that is designed to produce general answers.

10.1 Most General Answer

Generality can be defined in terms of the subsumption relation. Clauses with more specific information subsume those that are more general. Thus specifics subsume generics, which subsume hypotheticals. According to this rubric, a hypothetical is the most general answer, a generic is less general and a specific is not general. The most general answer [need acronym?] can be defined as follows: the conjunction of the hypothetical answers that are neither subsumed by other hypotheticals nor subsumed by generics, the generic answers that do not subsume any other generic answers, and the specific

answers that do not subsume any other answers. The non-subsumed hypothetical answers that are also not subsumed by generics are those with two properties: first, their hypotheticals have not been discharged (they are not subsumed by generics), and second, they are the most specific hypotheticals (they are not subsumed by other hypotheticals). While this is a most general answer, it is critical that any answer reflect what is known, and the most specific hypothetical answer does this better than a more general hypothetical which it subsumes. Hypotheticals that are part of the most general answer indicate a true information deficiency as described above. Generic answers that do not subsume other generic answers are those at the "top" of the hierarchy in terms of generality. Specific answers that do not subsume any other answers are the most general answers possible given the lack of generic answers.

10.2 Most Specific Answer

Similarly, a most specific answer can be defined as the conjunction of the following: the hypothetical answers that are neither subsumed by other hypotheticals nor subsumed by generics, the generic answers that are not subsumed by any other answers, and the specific answers. The criteria for including hypothetical answers is the same as for the most general answer. This reflects the fact that such answers reflect a lack of information, and the hypothetical answers included will be the most specific characterizations of that lack. When a generic answer is subsumed by another answer it means there is more specific information available, and it should not be included. Finally, specific answers are never subsumed.

A desired answer may be neither the most nor the least specific. It is not possible to determine the most or least specific answer until all resolvants descended from the query have been generated. This process may not terminate in case function symbols are included in the clauses.

If answers are produced as they are generated, which in some circumstances can be helpful and illuminating, it must be done with the knowledge that early answers may quickly be subsumed by later answers, which may lead to misunderstandings about how the information in the rule base is related. For example, if you have as answers, Chafic would like to eat a pastry, and Chafic would like to eat a napolean, you might have no idea that napoleans are pastry.

11 Summary

We have proposed a general characterization of answers in a rule base that extends the current notion of answer, and provides a framework for understanding previous types of answers. We have shown that all clauses descended from the query clause are indeed answers, despite the fact that many of them are disregarded by current systems, particularly Prolog and most theorem provers as well as many database systems.

In recognizing the importance of hypothetical answers, a new search strategy that focuses on "discharging" the hypothetical portion of answers has been proposed. This strategy has been employed for small problems, and larger experiments comparing it to other common search strategies are planned.

We have drawn attention to the fact that "answer" is by and large still identified with the "specific answer" proposed so long ago by Cordell Green [ref], despite the broadening of the definition by other researchers.

References

[1] Debra T. Burhans and Stuart C. Shapiro. Finding Answers with a Resolution Theorem Prover. In *AAAI-F99 Symposium on Question Answering*, Cambridge MA, 1999. AAAI Press.

[2] L. Cholvy and R. Demolombe. Querying a Rule Base. In *Proceedings of the First International Workshop on Expert Database Systems*, pages 365–371, 1986.

[3] Laurence Cholvy. Answering Queries Addressed to a Rule Base. *Revue d'Intelligence Artificielle*, 4(1), 1990.

[4] Matt Ginsberg. *Essentials of Artificial Intelligence*. Morgan Kaufmann Publishers, 1993.

[5] Cordell Green. Applications of theorem proving to problem solving. In Donald E. Walker and Lewis M. Norton, editors, *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 219–239. IJCAI, IJCAI, May 1969.

[6] Cordell Green. Theorem-proving by resolution as a basis for question-answering systems. In D. Michie and B. Melzer, editors, *Machine Intelligence 4*, pages 183–205. Edinburgh University Press, 1969.

[7] Amihai Motro. Using integrity contstraints to provide intensional answers to relational queries. In *Proceedings of VLDB89, the 15th International Conference on Very Large Databases*, pages 237–246, Amsterdam The Netherlands, August 1989. VLDB89.

[8] Amihai Motro. Intensional answers to database queries. *IEEE Transactions on Knowledge and Data Engineering*, 6(3):444–454, June 1994.

[9] Amihai Motro. Panorama: A Database System that Annotates Its Answers to Queries with their Properties . *Journal of Intelligent Information Systems*, 7:1–25, 1996.

[10] David Poole, Alan Mackworth, and Randy Goebel. *Computational Intelligence A Logical Approach*. Oxford University Press, 1998.

[11] Elaine Rich and Kevin Knight. *Artificial Intelligence*. McGraw Hill, 2 edition, 1991.

[12] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.