Partition Semantics and Cooperative Response

Kjell Johan Sæbø

Germanistisk institutt
Universitetet i Oslo
Postboks 1004
N- 0315 Oslo 3
kjelljs@hedda.uio.no

The paper is based on research in a project financed by the Norwegian Research Council for Science and the Humanities under a Programme for Information Technology in the Humanities.

ABSTRACT

This paper presents an attempt to construct an algorithm for the pragmatics of answers. It is desirable to anchor cooperative response, a centerpiece of artificial intelligence and computational linguistics, in the Gricean Cooperative Principle. This is possible if the conversational maxims are formalized along the lines of Groenendijk and Stokhof (1984). The paper describes a test application of their theory of question-answering in the form of a Prolog database query program, and discusses the implications for man-machine dialogue.

1. Introduction

Cooperative response is a key problem in work on natural-language interfaced database query systems. Computer experts tend to answer queries uncooperatively. In particular, they are rarely capable of exploiting the available knowledge in full. As soon as they improve with respect to the quantity of information offered, they risk offering insincere, irrelevant, or superfluous information.

The problem of cooperative question-answering can be approached from two different angles. Research has tended to focus on over-answering [1] - the tendency, especially pronounced in connection with yes-no questions, in humans to provide more information than explicitly requested. By contrast, the present study reverses the perspective and centers on partial response - a form of «under-answering», making the best of insufficient information in reaction to wh- questions.

A primitive system will not react properly to information states between zero information - the empty answer - and total information - some complete answer. On the one hand, it will respond «I don't know» even if it does have information, or on the other hand, it will render a complete answer even if it lacks total information. Thus it will represent its knowledge either too weakly or too strongly: It will violate either the maxim of Quantity or that of Quality in Gricean pragmatics.

The obvious point of departure for an algorithmic pragmatics of answers is the question theory of Groenendijk and Stokhof (1984). Grice's conversational maxims can be formalized with respect to a situation of utterance that defines a conversational background. And in this theory, the situation of question and answer does just that: The meaning of a question defines a set of propositions. Indeed, this semantics was partly motivated by a plan to formalize the Gricean framework. The semantics of questions is supplemented by a pragmatics of answers.

It has been an important goal to enable the system to provide as many different answers as possible in order to comply with the maxim of Quantity, and to select a concise response from among equivalent responses in order to obey the maxim of Manner. The paper concludes that it is hardly feasible to apply Quantity to machine question-answering uncompromisingly but that a system can and must be required to indicate whether or not the answer is complete.

Sections 2 and 3 set out the semantics of questions and pragmatics of answers. Sections 4 and 5 describe the implementation of the theory. It is shown that

- the Maxim of Quantity implies that there is only one correct answer
- that to detect that answer a system depends on "de dicto" information
- that it needs three truth values (partial response presupposes partial logic)
- that a "solution-driven" approach is preferable to an "answer-driven" one
- that the Maxim of Manner implies a sentence with a three-valued semantics
- that strict ("exhaustive") readings of term responses can be accommodated.

---

[1] cf. Lehnert (1977), Kaplan (1979), Hirschberg (1985), Sæbø (1988).

2. The Semantics of Questions

This section contains a brief sketch of the semantic theory of direct questions contained in *Studies on the Semantics of Questions and the Pragmatics of Answers* (Groenendijk and Stokhof 1984). The crucial point is that the sense of a question can be identified with a set of mutually disjunct propositions. If a proposition is the characteristic function of a set of indices, a question meaning is a function from indices to propositions, defining a partition of the set of indices.

Assuming a proposition is rendered as	• $\lambda i[\phi(i)]$
the sense of a yes-no question is rendered as	• $\lambda i \lambda j[\phi(j) = \phi(i)]$
and the sense of a wh- question is rendered as	• $\lambda i \lambda j[\{x \mid \phi\}(j) = \{x \mid \phi\}(i)]$.

For instance, the sense of «Who walks?» at an index results in the proposition that those who in fact walk walk - a complete survey of the walking ones at the index. The index j must yield the same value for the property as the index i.

A question's meaning is a relation between two indices. Defined via identity, it is an equivalence relation with equivalence classes: A question Q defines a partition I / Q of the index set I. It is convenient to represent the partition through a diagram. The below set of mutually disjunct propositions could stand for the meaning of «Which schooners are wrecked?» on the assumption that there are two schooners, Fortuna and Felicia. Wrt. the above formula, the set of j is the box containing i.

I / Q = Which schooners are wrecked?

Fortuna is wrecked but Felicia is not wrecked
Felicia is wrecked but Fortuna is not wrecked
Fortuna and Felicia are wrecked
Neither Fortuna nor Felicia is wrecked

Per definitionem, the elements of the partition are the complete answers to Q.

- P is a complete answer to Q iff $P \in I / Q$.
- P ist a partial answer to Q iff $P \neq \emptyset$ and $\exists X \subset I / Q : P = \cup X$.

Every complete answer is a partial answer in this wide sense. A partial answer may be termed an answer. How many answers are there? If the objects instantiating x - the wh- pronoun, as it were - are termed solutions and the number of solutions is n, the number of answers is 2^{2n} minus 2 (one for $P = \emptyset$ and one for $X = I / Q$).

3. The Pragmatics of Answers

To evaluate an utterance of a declarative sentence in Gricean pragmatics, two parameters are necessary: A conversational background and a state of information, corresponding to the notions **current exchange** and **evidence** in Grice's theory. For the declarative sentence to be compared with the conversational background and the state of information, Grice uses the term **contribution**. In the following reformulation of his **Cooperative Principle**, a sentence set serves this purpose.

> A sentence set S with respect to an information state B and a conversational background C is adequate iff it satisfies the four maxims in terms of B and C.

The set of propositions "S" corresponding to the sentences of S can be defined straightforwardly. Likewise, it is unproblematic to regard the state of information B as a set of propositions. The problem is defining the conversational background C. It can be solved, however, in the context of a response on the basis of the theory of Groenendijk and Stokhof - by identifying the conversational background with the set of mutually disjunct propositions defined by the question. As a result, all four maxims can be given a precise interpretation.

A response is irrelevant if it fails to distinguish among possible cases - if it is not **eliminative** (Manor 1982) for any **alternative** - if it does not rule out a single complete answer. An exception must be made: As long as it is somehow made clear that the Cooperative Principle remains active, the tautology shall count as relevant.

Relation. $\exists \varphi \in C : \cap\text{"S"} \cap \varphi = \emptyset$; excepting S = { *I don't know* } etc.

("S" is the set of propositions coming from the set of sentences S.)

The Maxim of Quantity is violated if an alternative such that evidence justifies its elimination stays open, ie. a response is under-informative if it is less eliminative than warranted by the information state. Technically, every alternative incompatible with the information state should be incompatible with the response too.

Quantity. $\forall \varphi \in C : \cap B \cap \varphi = \emptyset \supset \cap\text{"S"} \cap \varphi = \emptyset$.
Quality. $\forall \varphi \in C : \cap B \cap \varphi \neq \emptyset \supset \cap\text{"S"} \cap \varphi \neq \emptyset$.

These two maxims are perfect mirror images of each other. In this picture, Quality means that every alternative incompatible with the response should be incompatible with the information state as well.

The Maxim of Manner does not concern what is said but how it is said. This is why the response is not defined as a proposition but as a set of sentences. It should be free of redundancies. The formalization expresses that any response eliminating the same set of alternatives as the one under consideration should not have a lower cardinality. To be sure, this is problematic if the number of sentences is not finite.

Manner. $\forall S' : \{\varphi \mid \cap\text{"S"} \cap \varphi = \emptyset\} = \{\varphi \mid \cap\text{"S'"} \cap \varphi = \emptyset\} \supset |S| \leq |S'|$.

4. The Query and the Knowledge

The given pragmatics of answers is an extremely abstract theory: The separate modules act as filters on randomly generated responses. For an implementation, the relationship between query and response must be algorithmic. Note the distinction between answer and response: On the theory, an answer is defined in the abstract, while a response is what is actually said, to be compared with the set of answers defined by the question. Now it would be an unwise strategy to go directly to the answers as defined by the question and to try to verify - or **isolate** - or falsify - or **eliminate** - complete answers. This strategy would entail much double work: If you fail to prove one complete answer and return to zero to try to prove the next, you lose information you may have assembled. The same applies to the opposite strategy of successively trying to prove minimal answers. The point is that answers - complete or just partial - are not expressed in one fact each in the knowledge base of any system. For example, the answer that only Fortuna is wrecked will not be represented as one fact.

The knowledge-based question-answering system SHIPBASE represents an attempt to make cooperative response deterministic on the basis of the theory of Groenendijk and Stokhof. The challenge is to adjust to the two maxims Quantity and Manner. It is a primary goal to exploit as many options of partial response as possible. At the same time, the output should be kept as compact as possible. For the conception of a sensible system, this raises interesting questions.

Let us reflect for a moment on the implications of the Maxim of Quantity. The need to make the best of evidence that does not justify a complete answer involves a strict interpretation of the phrase «I do not know». To license the zero answer, the lack of **de re** knowledge will not suffice. Regarding a wh- question, one may be unable to answer the corresponding yes-no question with respect to any solution. Still, one may be able to eliminate alternatives on account of **de dicto** knowledge, like: There is a positive solution, there is a negative one, every solution is positive or none is. In particular, the piece of information that there is some positive solution confirms the **presupposition** of the question, which is something more than what «I do not know» conveys.

In designing the system, it is important not to simplify the task by tailoring the database and the inference engine to the requirements of cooperative response but to keep them natural and simple. Ideally, there should be nothing in the database that cannot be expressed in a simple sentence. Specifically, in the actual database there is no negation, there are no connectives but conjunction and no quantifiers but $\exists$, and there may be several representations of one fact. A fairly fine-grained inference machine plays a prominent rôle in deriving negative and complex facts.

The system treats wh- queries of various kinds uniformly:

«Which...», «How...», «What kind of...», «When...», and «Where...».

From a semantic point of view, there are three types, according as how the **solutions** are characterized. A solution is a possible instance of the wh- pronoun.

The strategy of trying to verify or falsify complete answers in succession - ie. trying to prove complete or minimal answers - could be called answer-driven. It is less efficient than one of trying to verify or falsify sentences about one solution at a time and to compose responses step by step, which could be called solution-driven. In the next section, we address the semantics of the various **response sentences** and their capacity with regard to partiality. First, however, we examine the possible **answers** in the abstract and their possible **sources** in the knowledge base.
For this purpose, we construct a simple example: «Which schooners are wrecked?» on the assumption that there are two schooners, Fortuna and Felicia.
There are 15 possible properly different answers, 4 complete ones, 10 partial ones, and the zero option («I don't know»). The 16th option would be the contradiction, eliminating all 4 partition elements, henceforth **alternatives**. A complete response eliminates 3 alternatives, a partial one just two or even just one.

Fortuna is wrecked but Felicia is not wrecked
Felicia is wrecked but Fortuna is not wrecked
Fortuna and Felicia are wrecked
Neither Fortuna nor Felicia is wrecked

Assume that the alternatives are numbered 1 through 4 from above in the diagram. The complete answers each coincide with one alternative 1, 2, 3, or 4. The sources of these answers in the knowledge base are on the one hand simple statements like: Fortuna is wrecked - Felicia is lost by being wrecked:

```
fact(vb(wrecked),sb(fortuna),ob(<loc(ex1),tem(ex2)>)) ->;
fact(vb(lost),sb(felicia),ob(<mod(wrecked),...>)) ->;
```

On the other hand, there are incompatible positive statements licensing the inference that a ship isn't wrecked, eg.: Fortuna has foundered - Felicia is lost by foundering:

```
fact(vb(foundered),sb(fortuna),ob(<loc(ex1),tem(ex2)>)) ->;
fact(vb(lost),sb(felicia),ob(<mod(foundered),...>)) ->;
```

The next stage concerns the minimally incomplete answers, first, the answer where the alternatives 3 and 4 are ruled out and the alternatives 1 and 2 are singled out. The sources are two positive statements about on the one hand wrecking and on the other hand eg. foundering, with unknown entities as subjects - ie. two existentially quantified statements - with additional facts that the same entities are schooners:

```
fact(vb(schooner),sb(ex3)) ->;
fact(vb(sailingship),sb(ex4),ob(<cat(schooner)>)) ->;
fact(vb(wrecked),sb(ex3),ob(<loc(ex1),tem(ex2)>)) ->;
fact(vb(lost),sb(ex4),ob(<mod(foundered),...>)) ->;
```

Conversely, the sources for the answer where the alternatives 1 and 2 are ruled out are, again, two existentially quantified statements, only that the quantification now does not affect names but verbs, or rather, ways of being lost: Fortuna and Felicia are lost in the same way. Then, it is clear that the two behave alike with respect to being wrecked, if Fortuna is then so is Felicia and conversely:

```
fact(vb(lost),sb(fortuna),ob(<mod(ex5),...>)) ->;
fact(vb(lost),sb(felicia),ob(<...,mod(ex5)>)) ->;
```

The source for isolating alternatives 1 and 4 or 2 and 4 is that Felicia isn't wrecked, via a corresponding positive statement, eg.:

```
fact(vb(lost),sb(felicia),ob(<mod(foundered),...>)) ->;
```

The last two of the six two-to-two answers coincide with the union of alternatives 1 and 3 or 2 and 3. They correspond to just the fact that Fortuna is wrecked or that Felicia is. Finally, regarding the minimal answers, the one that coincides with the complement of alternative 4 or 3 is derivable from the fact that there is a schooner wrecked or not wrecked, respectively.

We note that although the exemplified database facts are sufficient for deriving the answers according to Quality, they are not in general sufficient for deriving the answers according to Quantity. For instance, you have not yet arrived at the answer once you have been able to verify the union of alternatives 1 and 3 on the basis of the simple fact that Fortuna is wrecked. In addition, you have to have tried to verify or falsify both 1 and 3 and to have failed. So in terms of a cooperative response, the source of the answer 1-or-3 is on the one hand a fact that Fortuna is wrecked and on the other hand the **absence** of a relevant fact about Felicia. For an answer to be quantitatively "true", it must be the one and only qualitatively true answer in comparison with stronger ones, ie. every stronger answer must be weakly false. So for a system to cooperate wrt. Quantity, it must discriminate between failure and falsity, between a partial and a complete response.

In fact, it can be shown that with respect to a set of alternatives and a state of information, there is exactly one qualitatively and quantitatively appropriate answer.

We also note that an answer-driven approach entails unnecessary work. Each complete answer is a complex proposition, and if one cannot be proven one way or the other because one conjunct cannot be, the positive or negative information on another conjunct is not accumulated. A solution-driven approach, however, makes it possible to eliminate two alternatives at the same time. Two complete answers are merged into a simple proposition by treating the wh- question as a yes-no question wrt. a solution.

The next section discusses a set of response sentences and their semantics.

5. And the Response

For the special case of 2 solutions [2], the sentences (really, the sentence types) that can combine to form the response to a query are the following.

(1)	a/b	(5)	if any a/b
(2)	a and b	(6)	a as b
(3)	nil	(7)	some
(4)	...	(8)	some not

There is one syntactical constraint in this picture - (1) and (2) cannot cooccur. If we add that the sentences are ordered and that any sentence occurs only once, there are still a vast number (in fact, 2^{10}) of possible combinations, and intuitively, many of these responses are contradictory, and many have redundant parts. Ideally, there should be no more responses than answers, so much depends on introducing semantical and pragmatical constraints on cooccurrences into the meanings.

(1) a / b

This sentence has the semantics «a (b) is a positive solution».
The rule succeeds only once.

(2) and b

This sentence has the semantics «b is a positive solution too».
The rule succeeds iff the rule for (1) succeeds wrt. a and b is a positive solution.

(3) nil

This sentence has the semantics «every solution is negative».
The rule succeeds iff both the rule for (1) and the rule for (4) fail.

(4) ...

This sentence has the semantics «there is a neither positive nor negative solution».
The rule succeeds iff either a or b is neither positive nor negative.

(5) if any a / b

This sentence has the semantics «b (a) is a negative solution». The rule succeeds iff the rule for (4) succeeds wrt. a (or b) and b (or a) is a negative solution.

(6) a as b

---

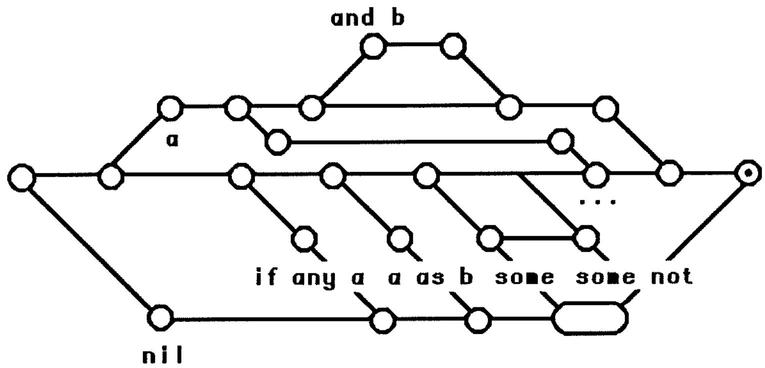
[2] The system is adapted to the general case of n solutions.

This sentence has the semantics «a is a positive solution and b is a positive solution or a is a negative solution and b is a negative solution» and the rule succeeds iff the rule for (4) does and it can be inferred that the two solutions have the same value.

```
(7)    some          (8)    some not
```

These two sentences have the semantics «there is an anonymous positive (negative) solution x». The rules succeed only if x is necessarily different from any familiar positive (or negative) solution and there is a familiar solution from which x is not necessarily different.

In procedural terms, the conditions correspond to the following chart.



Positive solutions are possible complete answers, and if such succeed without subsequent success of «...», then every complete answer is captured except «nil». If the rule for «a» or «b» and the rule for «...» both fail, that for «nil» automatically succeeds. Thus complete answers are not generated through success of positive or negative solutions but through success of positive solutions and failure of «...» or through failure of both.

Let us correlate the sentence meanings with the answers in section 5.
Sentence (1) **alone** isolates the complete answer 1 or the complete answer 2, and sentences (1) and (2) together isolate the complete answer 3. Sentence (3) isolates the complete answer 4. As for the partial answers, sentences (1) and (4) **together** isolate the union of 1 and 3 or the union of 2 and 3, sentence (5) isolates the union of 1 and 4 or the union of 2 and 4, sentences (7) and (8) together isolate the union of 1 and 2, and sentence (6) isolates the union of 3 and 4. Finally, the complement of 3 is isolated by sentence (7) and the complement of 4 is isolated by sentence (8).

There could exist a sentence type to eliminate just alternative 1 or alternative 2: Unidirectional implication (sentence (6) expresses bidirectional implication).

```
if a/b then b/a
```

But without disjunction in the knowledge base, there is no source there from which to derive this content. So two of the four minimal answers are not captured.

Otherwise, though, the Maxim of Quantity is secured since every answer is expressed by a possible response and every response sentence is subject to proof. And every response sentence except «...» (which on its own means «no evidence for any answer whatever») is eliminative, so any possible response is **relevant**. Although a response may not comply with the Maxim of Manner, much surplus is blocked because the success of one of the sentences (5)-(8) depends on the success of the sentence (4). These sentences only express partial answers and are - at best - redundant whenever a complete answer is valid. And sentence (4) says that none is.

Sentence (4) has a special semantics. It is not a strengthener but a **weakener**. This sign is not eliminative, ie. in terms of a two-valued semantics it is tautologous. Actually, it serves two functions: If its is the only succeeding rule, it signals that no elimination succeeds and so it expresses the zero answer. And, if the rule for «a» or «b» succeeds too, it signals that this sign is not to be interpreted exhaustively.

According to the Exhaustivity Hypothesis, the bare term response «a» means that «all other individuals...do not have the property» (Groenendijk and Stokhof 1984: 371). Thus «Fortuna» alone means that Felicia is not wrecked. If the open reading is intended, it must be explicitly indicated. Now the sentence «...» means that there is at least one indeterminate solution. The semantics is non-compositional: On its own, «...» means the tautology («I don't know»), but together with a term it means that not every other solution is negative. So it retracts part of «Fortuna».

We recall that for a partial response to be quantitatively correct, the complete answers that unify to form it must each be impossible to verify or falsify, so a third truth value is necessary to simulate anyhow. The notion of partial response presupposes an open world assumption. To be accurate, it presupposes a partial semantics with both a strong and a weak negation. This is necessary to have a choice between «no» and «I don't know» to a yes-no question. Without a distinction falsity-failure, any system systematically violates either the Maxim of Quality or that of Quantity. However, this distinction may be implicitly or explicitly represented, and «...» explicates the failure to verify or falsify a proposition.

Note that it is not the Exhaustivity Hypothesis as such that necessitates a rule like the rule for «...». In theory, we could adopt the converse convention: The bare term response «a» just means the partial answer and the complete answer must be indicated by adding «and none other». Then to prove «a and none other» you must disprove «...». Or, we could adopt the convention that any response is in principle partial and both positive and negative solutions are enumerated in the response. Unmentioned solutions are understood to be indeterminate. The response may happen to be complete - in case every solution is determinate. But the system will never know the difference. The point is that you need a content in the spirit of (4) once you want to express a complete answer via a general (non-)statement, ie. to draw an explicit distinction between a complete and a partial response.

6. Conclusions

The framework is super-exponential: If n is the number of solutions, there are 2^n complete answers (alternatives) and 2^{2n} (partial) answers. For instance, if there are 10 solutions there are 2^{20} answers (÷ 1 for the contradiction). In any approximately realistic application, this must affect time performance fatally. It could be concluded that the framework is flawed. Indeed, in a case of a hundred solutions where none is provably positive or negative it would be absurd to start responding minimally. It seems more rational then to restrict oneself to answers that derive from positive or negative familiar solutions or fail to eliminate altogether, disregarding intermediate information.

It is worth noting that natural-language dialogue is customarily confined to a tiny universe of discourse. As soon as domains grow large, another kind of query is possibly involved, typical of database queries. As far as this kind is to be natural-language interfaced, it may have to submit to natural-language constraints regarding "aboutness" (Stechow & Zimmermann 1984: 12) as well. In short, it might be that query and response is different from question and answer, and that only the latter is sensibly treated partitionally.

But if a system restricts itself to answers based on **de re** information - proven information about specific solutions - still the theory of Groenendijk and Stokhof has one essential implication. The system should be required to indicate whether or not the response is complete. To this end, it must be enabled to draw the distinction between complete and only partial answers.

This is particularly true if the response sentences are all, as is often the case in human as in machine dialogue, positive solution terms. For one thing, if the system is unaware of the exhaustive nature of term responses, it will respond just complete answers and in many cases lie - violate the Maxim of Quality. Or, if it is aware of this property of term responses but unable to recognize a complete answer once it sees one, it will respond just partial answers and in many cases withhold information - violate the Maxim of Quantity. (Mutatis Mutandis, the same applies without the Exhaustivity Hypothesis.)

So a system should be equipped to know the difference between completeness and partiality in answers, which amounts to knowing the meaning of a sentence like

«and maybe more», or a sentence like «and none other».

This meaning is, of course, something like

'there is (not) a solution such that it is not positive and it is not negative'.

And the cost of installing this capacity in a system is moderate. SHIPBASE has it in the form of the truth condition of the response sentence «...». This sign may also be paraphrased **this is no complete answer**, and the program demonstrates the possibility of treating this non-compositional content in an algorithmic fashion.

REFERENCES

Fenstad, J. E. / E. Vestre (1988). «Representing Direct Questions» (= COSMOS-Report 04). Department of Mathematics, University of Oslo.

Groenendijk, J. / M. Stokhof (1984). *Studies on the Semantics of Questions and the Pragmatics of Answers*. Dissertation, University of Amsterdam.

Hirschberg, J. B. (1985). *A Theory of Scalar Implicature*. Dissertation, University of Pennsylvania.

Kaplan, S. J. (1979). *Cooperative Responses from a Portable Natural Language Data Base Query System*. Dissertation, University of Pennsylvania.

Lehnert, W. (1977). «A Conceptual Theory of Question Answering», in Proceedings of IJCAI '77.

Lloyd's Weekly Shipping Index 1900. «Maritime Losses and Casualties».

Manor, R. (1982). «Answers and Other Reactions», in Theoretical Linguistics 9, 69-94.

Sæbø, K. J. (1988). «A Cooperative Yes-No Query System Featuring Discourse Particles», in Proceedings of COLING '88, Budapest.

Stechow, A. v. / T. "E." Zimmermann (1984). «Term Answers and Contextual Change», in Linguistics 22, 3-40.