Active Logic Applied to Cancellation of Gricean Implicature*

Khemdut Purang and Don Perlis
Department of Computer Science
University of Maryland
College Park, MD 20742
kpurang@cs.umd.edu and perlis@cs.umd.edu

John Gurney
Software and Intelligent Systems Division
Army Research Laboratory, Adelphi MD 20783
gurney@assb01.arl.mil

Abstract

Using a simple example of a dialog with an implicature that arises part way through and then is later retracted, we discuss how Gricean maxims and nonmonotonicity may relate to each other and to a computational treatment of implicature. In effect we seek to track reasoning along Gricean lines over time. We present our own computational approach to this, giving an implementation in the formalism of active logics.

Introduction

In the years since the appearance of Grice's (Grice 1967), the theory of implicature has been widely discussed by philosophers, linguists, cognitive scientists and others. The early articulation of the theory was accomplished by Grice himself (Grice 1975; 1978; 1981). Many others have developed the theory (Gazdar 1979; Horn 1972; Levinson 1983), assumed it (Kripke 1977), modified it (Sperber & Wilson 1986), and criticized it (McCafferty 1987; Thomason 1990).

We think that most of what Grice proposed is extremely insightful and that we should continue to re–examine his examples, observations, and theoretical proposals. The theory is simple and elegant. It preserves two–valued logic and a place for the well–developed machinery of deductive inference. It has wide coverage; many seemingly disparate phenomena are explained. And many of the classic examples of implicature are plausible. However, Grice's theory has not produced widely used computational implementations. Nevertheless, we think that it would be hard to imagine how one would implement a computational theory of implicature if one were to actually reject very much of what Grice proposes.

In this paper we report on our implementation of a small portion of the computational project. We are interested here primarily in Grice's observation that conversational implicatures are cancelable. If one accepts Grice's analysis of conversations, there are at least five ways that the hearer might process any given utterance of the speaker. The hearer might come to believe that the speaker is (1) observing the maxims, (2) simply violating the maxims, (3) opting out of the conversation, (4) faced with a clash among maxims, or (5) flouting (usually one of) the maxims. Of course, deciding which of these cases is in play is one of the open problems for the computational project. Here we will simply assume case (1). So the hearer assumes that the speaker is observing the maxims (of Quality and Relevance in our examples).

We wish to seriously consider how cancellation might work and how to implement the actual positing and withdrawal of implicatures *in real time*. In this paper we show that the same underlying framework of active logic that we earlier (Gurney, Perlis, & Purang 1995) applied to *presuppositional* inference in real–time (evolving) dialog–processing also is applicable to inference of *implicatures*. In our first example we model this as a particular kind of non–monotonic inference in active logic, using an example based on our earlier work with presuppositions.

Active Logic

Active logic (Elgot-Drapkin & Perlis 1990; Miller & Perlis 1993) is a family of formalisms developed for the purpose of modeling the reasoning process in a way that respects the passage of time as reasoning proceeds. These formalisms have been applied to a number of domains, from multi–agent interaction to deadline–coupled planning, from fully–decidable default reasoning to reasoning in the presence of contradictions, from correcting misidentification errors to perceptual reference.

Rather than proceeding from one nonmonotonic theory (with one set of axioms) to another nonmonotonic theory (with an updated set of axioms) there is one evolving theory in active logic. It models a process of thinking that takes a reasoner from one belief state to the next. As a default everything believed at step n would be inherited to step n + 1. But there are various

*This work was supported in part by the Army Research Laboratory through a grant from the Army Research Office and by the National Science Foundation. We thank Betsy Klipple and Michael Morreau for discussion of these issues.

rules that modify this blanket inheritance. For example, if p and not p appear at step n then the belief contra(p, not p) appears at step n + 1. Then both p and not p are blocked from inheriting to step n + 2. For our present task, this is perhaps the most important characteristic on active logic. It works by forward chaining from step to step allowing contradictions to appear as they will. It uses detection of explicit contradictions to disinherit propositions from the belief set. In this way active logic achieves some of the effects of various nonmonotonic logics but in a different way.

In this paper we will discuss in some detail the application of active logic to two short dialogs that yield answers to yes–no questions by implicature.

First Example

(A) Kathy: **Are the roses fresh?**
(B) Bill: **They are in the fridge.**
(C) Bill: **But they're not fresh.**

We will model Kathy as the hearer who is trying to find an answer to her question by thinking about the import or relevance of Bill's utterances. At (B) it looks like Bill has given an indirect answer to Kathy's question. Kathy assumes what Bill said is relevant; so she thinks that knowledge about refrigerators keeping things fresh is relevant. Given this Kathy can draw the inference that the roses are fresh. This is the implicature. But then at (C), we presume, Bill's second utterance undoes the implicature.

It is this cancellation of a previous implicature that we seek to capture computationally. Note that if Bill's second utterance had been made first, there never would have arisen an implicature that the roses are fresh. This non–commutativity shows up only when we take time–evolution of dialog into account: the final state of implicature of the entire dialog is the same either way, but in the original version the hearer (Kathy) first goes through an intermediate state of accepting an implicature that she later gives up. This is a genuine part of dialog understanding that any real agent (human or machine) must go through during dialog. This is what we mean by cancellation of an implicature. In the dialog (A), (B), (C) there is cancellation. In the dialog (A), (C), (B) there is no cancellation.

Active Logic Implementation of First Example

Our implementation in active logic consists of (a) rules representing three kinds of knowledge and (b) an inference procedure that applies these rules repeatedly taking us from one step to the next. The three kinds of knowledge are: the active logic meta–theory, general beliefs about dialogs involving questions and answers, and background beliefs about refrigerators, roses, food, and so on. Some of these rules appear in Figures 1, 2 and 3. The meta–theory rules say things like: *If there is a contradiction don't inherit either alternative*, *Otherwise inherit anything that you can*, and *Update the time by 1 from one step to the next*. The discourse rules say things like *Believe anything that the speaker informs you*, *Believe any direct response to your question*, and *Try to figure out what an indirect response to your question means*. The background beliefs say things like *Things in fridges are cold*, *Cold roses are fresh*, and *Things in fridges are edible*. These last three beliefs happen to be defeasible; the rules will only fire if the right hand sides of these rules cannot be proven false.

For our investigations we are not treating our examples (the one above and the other one below) as task–oriented dialogs in the way those dialogs studied by (Allen & Perrault 1980; Green 1990; Green & Carberry 1994) are task oriented. There are at least two ways that a dialog can involve intentions, goals, and plans of the participants. There are conversational, or Gricean, intentions and there may be non conversational intentions, such as the intention to catch a train or get ready for a prom. For our examples, as we are taking them, one need pay attention to only the first kind of intention. For the rest, Bill exploits, and Kathy must pay attention to, what (Green 1990) calls plan independent knowledge (about how refrigerators work, etc.) to convey his implicature. Kathy may not have to think anything about Bill's non conversational plans, goals or intentions.

For the purposes of this investigation we simply assume that Kathy has perceived the relevant conversational intentions. These lead her to have beliefs about the relevance of Bill's utterances to her question. When Bill says (B) **They are in the fridge** we simply assume Kathy comes to a belief that we represent as **utt:(respond(bill,kathy,infridge(r1), q_yes_no(fresh(r1)))@2)** This means something like: Bill responded to Kathy regarding the yes–no question by saying r1 are infridge. Since this is a response Kathy will, at a later step, think that infridge(r1) is relevant to her question. We represent this belief as **rel(fresh(r1)):infridge(r1)**. Knowing what is relevant then leads to searching the background beliefs for knowledge about things being in refrigerators. We will comment on some of this in the next section.

Output Trace for First Example

Our implementation in active logic produces the positing of an implicature followed by its cancellation as we move from (B) to (C). There are several steps between (B) and (C) and several more after (C). The output trace below shows that at step 9 (which occurs after the utterance (C)) a contradiction briefly appears. Then it is withdrawn. At this point Kathy has no belief at all whether or not the roses are fresh. Then the belief that they are not fresh is restored using a rule that expresses the maxim of Quality. Below we reproduce and comment in more detail on some of the output trace

```
% inheritance rule: we inherit anything that is not killed and is not
%     itself a kill

(mt:(((`(Q:X)) & (?(kill(X)))  & eval(\+ (Q = time)) & (?(k2(Q)))
      & eval(\+ (X = k2(_))) & eval(\+ (X = kill(_)))) => (Q:X)))@0.

%contradiction detection rule: when we detect a contradiction, we add
%     a contra belief at the next step and we kill both contradictands
%     and the contra belief itself (so these don't propagate)

(mt:((`(Q:X) & `(W:not(X)) & (?(kill(not(X))))) =>
    mt:(kill(not(X)))))@0.
(mt:((`(Q:X) & `(W:not(X)) & (?(kill(contra((Q:X), (W:not(X))))))) =>
    mt:(kill(contra((Q:X), (W:not(X)))))))@0.
(mt:((`(Q:X) & `(W:not(X)) & (?(kill(contra((Q:X), (W:not(X))))))) =>
    mt:(contra((Q:X), (W:not(X))))))@0.
(mt:((`(Q:X) & `(W:not(X)) & (?(kill(X)))) => mt:(kill(X))))@0.

% the clock rule

(mt: ((`(time:now(T)) & eval(T1 is T+1)) => time:(now(T1))))@0.
```

Figure 1: Active Logic Rules

for this example.

Step 1

```
time:now(1)
bel:roses(r1)
bel:whoami(kathy)
utt: (q_yes_no(kathy,bill,fresh(r1))@1)
```

Kathy asks (A) **Are the roses fresh?**

Step 2

```
utt: (q_yes_no(kathy,bill,fresh(r1))@1)
bel:whoami(kathy)
bel:roses(r1)
time:now(2)
utt: (respond(bill,kathy,infridge(r1),
   q_yes_no(fresh(r1)))@2
```

Bill says (B) **They are in the fridge**.

Step 3

```
rel(fresh(r1)):infridge(r1)
bel:infridge(r1)
time:now(3)
utt: (respond(bill,kathy,infridge(r1),
   q_yes_no(fresh(r1)))@2)
bel:roses(r1)
bel:whoami(kathy)
utt: (q_yes_no(kathy,bill,fresh(r1))@1)
```

Kathy starts to think about what Bill's response means. The proposition **rel(fresh(r1)):infridge(r1)** means some thing like: With regard to the question whether or not **fresh(r1)** the fact that **infridge(r1)** could be relevant. The active logic system, in the course of its normal inheritance procedure will try to fire any rules that it can using **infridge(r1)**. This is our simplified model for Kathy's thinking about or trying to figure out the relevance of Bill's indirect answer.

Step 4

```
utt: (q_yes_no(kathy,bill,fresh(r1))@1)
bel:whoami(kathy)
bel:roses(r1)
utt: (respond(bill,kathy,infridge(r1),
   q_yes_no(fresh(r1)))@2)
rel(fresh(r1)):infridge(r1)
time:now(4)
bel:infridge(r1)
rel(fresh(r1)):cold(r1)
rel(fresh(r1)):small(r1)
rel(fresh(r1)):dead(r1)
rel(fresh(r1)):edible(r1)
```

Nothing interesting yet. Kathy is still thinking. She considers the relevance of the temperature, size, and edibility of the roses.

Step 5

```
rel(fresh(r1)):fresh(r1)
time:now(5)
rel(fresh(r1)):edible(r1)
rel(fresh(r1)):dead(r1)
rel(fresh(r1)):small(r1)
rel(fresh(r1)):cold(r1)
bel:infridge(r1)
rel(fresh(r1)):infridge(r1)
```

```
% usually when X informs us of P, we believe P.

(bel: (( (inform(X, Y, P)@_) & whoami(Y) & (?(ab2(X, Y, P)))
         ) => (bel:P)))@0.

%(bel: (( (respond(X, Y, P, Q)@_) & whoami(Y) & (?(ab2(X, Y, P))) &
%         eval(\+ Q = q_yes_no(_))  ) => (bel:P)))@0.

% we always believe what people tell us.

(bel: (( (respond(X, Y, P, Q)@_) & whoami(Y) & (?(ab2(X, Y, P)))
           ) => (bel:P)))@0.

% indirect responses to yes-no questions- we try to figure what it means

(bel: (( (respond(_, X, P, q_yes_no(Q))@T) & whoami(X) & now(T) &
          eval(\+ P = Q) & eval(\+ P = not(Q)))
      => (rel(Q):P) ))@0.

% if we have figured the answer, we make it an implicature and we stop
% trying to find out what the response meant. In this case, we just lose
% all the irrelevant beliefs we came across.

(bel: (( `((rel(Q)):P) & (?(k2(rel(Q)))) & eval(P = Q)) => imp:P))@0.
(bel: (( `((rel(Q)):P) & (?(k2(rel(Q)))) & eval(P = Q)) =>
    mt:k2(rel(Q))  ))@0.
(bel: (( `((rel(Q)):P) & (?(k2(rel(Q)))) & eval(P = not(Q))) => imp:P))@0.
(bel: (( `((rel(Q)):P) & (?(
```

Figure 2: Discourse Rules

```
utt: (respond(bill,kathy,infridge(r1),
   q_yes_no(fresh(r1)))@2)
bel:roses(r1)
bel:whoami(kathy)
utt: (q_yes_no(kathy,bill,fresh(r1))@1)
```

Here at step 5 Kathy finds that the roses are fresh. That is the import of

`rel(fresh(r1)):fresh(r1).`

Inferring either this or

`rel(fresh(r1)):not(fresh(r1))}`

will stop Kathy's search for an answer to her yes–no question.

Step 7

```
time:now(7)
imp:fresh(r1)
bel:infridge(r1)
utt: (respond(bill,kathy,infridge(r1),
   q_yes_no(fresh(r1)))@2)
bel:roses(r1)
bel:whoami(kathy)
utt: (q_yes_no(kathy,bill,fresh(r1))@1)
utt: (respond(bill,kathy,not(fresh(r1)),
   q_yes_no(fresh(r1)))@7)
```

Now Bill says (C) **The roses are not fresh**.

Step 9

```
time:now(9)
mt:kill(fresh(r1))
mt:contra(imp:fresh(r1),
   bel:not(fresh(r1)))
mt:kill(contra(imp:fresh(r1),
   bel:not(fresh(r1))))
mt:kill(not(fresh(r1)))
bel:infridge(r1)
bel:not(fresh(r1))
imp:fresh(r1)
utt: (respond(bill,kathy,infridge(r1),
   q_yes_no(fresh(r1)))@2)
bel:roses(r1)
bel:whoami(kathy)
utt: (q_yes_no(kathy,bill,fresh(r1))@1)
utt: (respond(bill,kathy,not(fresh(r1)),
   q_yes_no(fresh(r1)))@7)
```

Kathy believes what Bill says, **bel:not(fresh(r1))** but now detects a contradiction. So neither the implicature **imp:fresh(r1)** nor this belief will inherit to the next step.

```
% things in fridges are cold

(bel: ((`(Z:infridge(X)) & (?(k2(Z))) & eval(Z = rel(_)) &
          (?(not(cold(X))))) => Z:cold(X)))@0.

% things in fridges are small

(bel: ((`(Z:infridge(X)) & (?(k2(Z))) & eval(Z = rel(_)) &
         (?(not(small(X))))) => Z:small(X)))@0.

% things in fridges are dead

(bel: ((`(Z:infridge(X)) & (?(k2(Z))) & eval(Z = rel(_)) &
       (?(not(dead(X))))) => Z:dead(X)))@0.

% things in fridges are edible

(bel: ((`(Z:infridge(X)) & (?(k2(Z))) & eval(Z = rel(_)) &
         (?(not(edible(X))))) =>  Z:edible(X)))@0.

% cold roses are fresh

(bel: ((roses(X) & `(Z:cold(X)) & eval(Z = rel(_)) &
       (?(k2(Z)))  & (?(not(fresh(X))))) => Z:fresh(X)))@0.

% old roses are not fresh.

(bel: ((`(Z:old(X)) & roses(X) & eval(Z = rel(_))
       & (?k2(Z)) & (?(fresh(X)))) => Z:not(fresh(X)) ))@0.
```

Figure 3: Background Beliefs

Step 10

```
utt: (respond(bill,kathy,not(fresh(r1)),
   q_yes_no(fresh(r1)))@7)
utt: (q_yes_no(kathy,bill,fresh(r1))@1)
bel:whoami(kathy)
bel:roses(r1)
utt: (respond(bill,kathy,infridge(r1),
   q_yes_no(fresh(r1)))@2)
time:now(10)
bel:not(fresh(r1))
bel:infridge(r1)
```

But the belief that the roses are not fresh is derivable from **utt:(inform(bill,kathy, not(fresh(r1)))@7)** at step 9 because Kathy believes everything Bill said. However, the implicature from (B), that the roses are fresh cannot be rederived; an utterance initiates a search for relevance and hence implicature only at the step it is perceived. In active logic we spread the reasoning over a sufficient number of steps to, so to speak, divide and conquer some of the complex thinking that happens during discourse understanding.

Second Example

(A) Kathy: **Are the roses fresh?**
(B) Bill: **They are in the fridge.**
(D) Bill: **But they are old.**

One of the criticisms of Grice is that his reconstructions of the hearer's reasoning are not sensitive to possible changes in context (McCafferty 1987). This second example shows that reasoning about implicatures can not only be sensitive to context; it can also be sensitive to changes in context induced by the conversation (provided that one or the other of the Gricean maxims is assumed to hold). After (B) we have the implicature that the roses are fresh, as in the first example. But (D) then changes the context of shared knowledge. Now the implicature should be withdrawn. Some discussions of cancellation rely on an explicit utterance that negates the implicature. McCafferty rightly observes that whether implicatures can be inferred at all depends on context. Using active logic we can account for this; if (D) were part of the contextual shared knowledge, then the implicature that the roses are fresh would not have been inferred after (B). But we can also account for the above example where the context actually changes during the conversation. Extending the previous terminology, we might call (D) an indirect cancellation.

Output Trace for Second Example

The common background includes the defeasible belief that old roses are not fresh (see Figure 3). We saw in the first example that the implicature at (B) was based on defeasible beliefs about things in fridges and about cold roses. This means that we could have a Nixon Diamond after (D); there will be two (defeasible) implicatures: one that the roses are fresh and the other that they are not fresh. This is what happens in our trace as we pick up Kathy's thinking at step 7.

Step 7

```
time:now(7)
imp:fresh(r1)
bel:infridge(r1)
utt: (respond(bill,kathy,infridge(r1),
   q_yes_no(fresh(r1)))@2)
bel:roses(r1)
bel:whoami(kathy)
utt: (q_yes_no(kathy,bill,fresh(r1))@1)
utt: (respond(bill,kathy,old(r1),
   q_yes_no(fresh(r1)))@7)
```

Here is where Bill says (D) **They are old**.

Step 8

```
utt: (respond(bill,kathy,old(r1),
   q_yes_no(fresh(r1)))@7)
utt: (q_yes_no(kathy,bill,fresh(r1))@1)
bel:whoami(kathy)
bel:roses(r1)
utt: (respond(bill,kathy,infridge(r1),
   q_yes_no(fresh(r1)))@2)
imp:fresh(r1)
time:now(8)
bel:old(r1)
bel:infridge(r1)
rel(fresh(r1)):old(r1)
```

As before, Kathy begins to think about the relevance to her question of Bill's utterance. So she now thinks that the age of the roses is relevant to her question **rel(fresh(r1)):old(r1)**. She will begin searching for what that relevance could mean.

Step 9

```
rel(fresh(r1)):not(fresh(r1))
time:now(9)
rel(fresh(r1)):old(r1)
bel:infridge(r1)
bel:old(r1)
imp:fresh(r1)
utt: (respond(bill,kathy,infridge(r1),
   q_yes_no(fresh(r1)))@2)
bel:roses(r1)
bel:whoami(kathy)
utt: (q_yes_no(kathy,bill,fresh(r1))@1)
utt: (respond(bill,kathy,old(r1),
   q_yes_no(fresh(r1)))@7)
```

She happens to discover the relevance after only one step. It is that the roses are not fresh **rel(fresh(r1)):not(fresh(r1))**.

Step 10

```
utt: (respond(bill,kathy,old(r1),
   q_yes_no(fresh(r1)))@7)
utt: (q_yes_no(kathy,bill,fresh(r1))@1)
bel:whoami(kathy)
bel:roses(r1)
utt: (respond(bill,kathy,infridge(r1),
   q_yes_no(fresh(r1)))@2)
imp:fresh(r1)
rel(fresh(r1)):old(r1)
mt:kill(not(fresh(r1)))
mt:kill(contra(imp:fresh(r1),
   rel(fresh(r1)):not(fresh(r1))))
mt:contra(imp:fresh(r1),
   rel(fresh(r1)):not(fresh(r1)))
mt:kill(fresh(r1))
time:now(10)
bel:old(r1)
bel:infridge(r1)
mt:k2(rel(fresh(r1)))
rel(fresh(r1)):not(fresh(r1))
```

Here the contradiction is discovered between the the implicature from (B) and the more recent implicature from (D). So both are marked **kill** so they will not be inherited at the next step.

Step 12

```
utt: (respond(bill,kathy,old(r1),
   q_yes_no(fresh(r1)))@7)
utt: (q_yes_no(kathy,bill,fresh(r1))@1)
bel:whoami(kathy)
bel:roses(r1)
utt: (respond(bill,kathy,infridge(r1),
   q_yes_no(fresh(r1)))@2)
time:now(12)
bel:old(r1)
bel:infridge(r1)
```

Here Kathy has reached a state where she cannot infer anything helpful about her original question. It worked out this way because Bill said things which only implicated that the roses were fresh or not fresh. But both implicatures were cancelled. Active logic *can* allow one to infer again what one has just withdrawn. But in the model we have designed the search for the relevance or import of an utterance only begins at the step immediately following the perception of the utterance as relevant to a question. This captures the fact that it is utterances, not beliefs, that give rise to implicatures. It also captures the idea that cancellation just eliminates the implicature; any new implicature would have to wait for a new utterance. However, we are not prepared to say that this strategy is generally applicable.

The Gricean Maxims

In this investigation, we have not represented any of the Gricean maxims explicitly. We have been regarding them as something like specifications for building a discourse participant. Each of our discourse rules (shown in Figure 2) articulates one or more of the maxims. The rules about believing the content of any utterance implement the maxim of quality. The rule dealing with indirect responses relates to both the maxim of quantity and the maxim of relevance. For it is this rule that produces the relevance beliefs like **rel(fresh(r1)):infridge(r1)**. These are the beliefs that begin a search for relevant rules to fire that may lead to an answer to one's question. In our model for the first example Kathy begins looking for and answer immediately after hearing (B) *They are in the fridge*. There is no waiting to hear what comes next. We can view this as following the maxim of quantity; Kathy assumes for the moment that Bill has said all that is relevant. But no harm was done, since implicatures can be cancelled.

The active logic rules shown in Figure 1 appear to be, in this context, rules for rational thought rather than rules specifically for understanding conversations.

Related Research

(Lascarides & Oberlander 1993) use a system of default rules to infer implicated temporal relations in discourses. One of their domain problems involves the conflict between rules that reflect the Gricean manner maxim and rules that reflect the Gricean relevance maxim. (They may or may not wish to characterize their rules in this way.) They employ what seem to be well motivated methods for prioritizing default rules in order to resolve conflicts. For the examples we have discussed our system also resolves some conflicts. Implicatures give way to literal assertions and conflicting implicatures lead to suspension of belief. It may be interesting that some of this behavior in our system emerges from a combination of other behaviours that embody Gricean maxims and principles of rationality. We have not yet investigated whether and how we would implement more explicit prioritization in active logic. There are, of course, important differences as to how cancellation works and how the effects of, say, prioritized default rules arise. As far as we can tell the methods under study by Lascarides, Oberlander, and Asher do not model the drawing and later cancellation of implicature as we do. Recalling our first example, the difference we emphasize between the dialog (A), (B), (C) and the dialog (A), (C), (B) is not important to theirs, as well as perhaps most other, nonmonotonic theories.

(Green & Carberry 1994) consider task oriented dialogs. Their treatment of yes–no question dialogs influenced our relevance rules in Figure 2. They and others use abductive inference to constrain the search space when trying to derive a direct answer from the utterance of an indirect answer. We think we could have used abductive inference for the same purpose in our system. We chose instead to work with a perhaps less efficient strategy. Given a relevant proposition one simply looks for rules in which that proposition can be instantiated. Then one generates inferences from those rules until a direct answer is found (if it is). This method may have some cognitive validity. Perhaps some indirect answers are more difficult to appreciate than others and perhaps that is because people only spend a short time trying out inferences that combine what was said with what they know. We have nothing more to say beyond these speculations at this time.

As far as active logic goes, we could attempt to subsume any or all of the above methodologies. That is, we could attempt to implement their methods in active logic. This would be an interesting project.

Conclusion

We offer active logic as an approach to use when modelling human dialogs. Our discussions of the two yes–no question dialogs illustrate how one might use this approach to model and study several of the computational aspects of conversation. We have used rules that are plausibly based on Gricean maxims as well as more general principles of rationality. This is a beginning effort which we intend to expand upon. This approach suggests various possibilities for taking time into account computationally. Our discourse rule for interpreting responses to questions (in Figure 2) is one example. This is the rule that initiates the search for an implicature of an utterance. But it can only be used once, at the step where the utterance first appears. This restriction, which refers explicitly to the current time step, is essential (in our model) for achieving the proper ultimate belief states.

We have implemented one kind of search process that Kathy uses to find the relevance of Bill's utterances to her question. And we have constrained that search by considering only yes–no questions and assuming that Kathy has already perceived that Bill's utterance is relevant to her question. These are obvious places where our models can be extended, either by subsuming existing solutions or by working with the unique machinery of active logic.

Scaling–up problems arise when a system that works well on a small set of beliefs ceases to work when more beliefs are added. Within the framework of yes–no question dialogs, we have addressed some of the problems that would arise in a more realistic or full–scale setting. We have: (1) included irrelevant rules along with methods for focusing the search on promising rules, (2) illustrated how conflicting defaults are handled in some cases, and (3) used discourse knowledge to focus on possible implicatures.

References

Allen, J., and Perrault, C. R. 1980. Analyzing intentions in dialogues. *Artificial Intelligence* 15.

Elgot-Drapkin, J., and Perlis, D. 1990. Reasoning situated in time I: Basic concepts. *Journal of Experimental and Theoretical Artificial Intelligence* 2(1):75–98.

Gazdar, G. 1979. *Pragmatics, Implicature, Presupposition and Logical Form.* New York: Academic Press.

Green, N., and Carberry, S. 1994. A hybrid reasoning model for indirect answers. Association for Computational Linguistics.

Green, N. 1990. Normal state implicature. Association for Computational Linguistics.

Grice, H. P. 1967. *Logic and Conversation.* William James Lectures. Harvard University.

Grice, H. P. 1975. *Syntax and Semantics 3: Speech Acts.* Academic Press. chapter Logic and Conversation.

Grice, H. P. 1978. *Syntax and Semantics 9: Pragmatics.* Academic Press. chapter Further Notes on Logic and Conversation.

Grice, H. P. 1981. *Radical Pragmatics.* Academic Press. chapter Presupposition and Conversational Implicature.

Gurney, J.; Perlis, D.; and Purang, K. 1995. Active logic and heim's rules for updating discourse context. IJCAI Workshop on Context in Natural Language Processing.

Horn, L. R. 1972. *On the Semantic Properties of Logical Operators in English.* Ph.D. Dissertation, UCLA.

Kripke, S. 1977. *Contemporary Perspectives in the Philosophy of Language.* University of Minnesota Press. chapter Speaker's Reference and Semantic Reference.

Lascarides, A., and Oberlander, J. 1993. Temporal coherence and defeasible knowledge. *Theoretical Linguistics* 19(19).

Levinson, S. C. 1983. *Pragmatics.* Cambridge University Press.

McCafferty, A. 1987. *Reasoning about Implicature: A Plan Based Approach.* Ph.D. Dissertation, University of Pittsburgh.

Miller, M., and Perlis, D. 1993. Presentations and this and that: logic in action. Cog-Sci.

Sperber, D., and Wilson, D. 1986. *Relevance: Communication and Cognition.* Harvard University Press.

Thomason, R. H. 1990. *Intentions in Communication.* MIT Press. chapter Accommodation, Meaning, and Implicature: Interdisciplinary Foundations for Pragmatics.