Applying Theories of Communicative Action in Generation Using Logic Programming

Matthew Stone
Department of Computer and Information Science
University of Pennsylvania
200 South 33rd St, Philadelphia PA 19104
matthew@linc.cis.upenn.edu

One of the most attractive arguments for taking a logical approach to communication, action and knowledge is the prospect of accounting for the ability of agents to perform inference in performing and recognizing communicative actions. In this squib, I illustrate how NLG systems can pose and evaluate logical queries inspired by the work of (Moore, 1985; Morgenstern, 1987; Davis, 1994) to assess the inferential effects of their communicative actions. I have developed a constraint logic programming language DIALUP designed to execute such queries efficiently.

Here I will use a multimodal logic with at least three S4 necessity operators: $[\mathrm{S}]$, representing the knowledge of the speaker; $[\mathrm{A}]$, representing the knowledge of the addressee; and $[\mathrm{CP}]$, representing their common knowledge. The eight formal rules governing these modalities, given below,[1] represent a reasonable idealization of conversation (see references above and (Clark, 1996) for shared knowledge):

$$[\mathrm{S}]p \supset p \quad [\mathrm{A}]p \supset p \quad [\mathrm{CP}]p \supset p$$
$$[\mathrm{S}]p \supset [\mathrm{S}][\mathrm{S}]p \quad [\mathrm{A}]p \supset [\mathrm{A}][\mathrm{A}]p \quad [\mathrm{CP}]p \supset [\mathrm{CP}][\mathrm{CP}]p$$
$$[\mathrm{CP}]p \supset [\mathrm{S}]p \quad [\mathrm{CP}]p \supset [\mathrm{A}]p$$

In this logic, formulas with nested implications provide a way to gauge the effects of actions. Let's start with the simplest example, an indirect answer, as described by (Green and Carberry, 1994). You are asked the question whether Q is true, and you want to give not only your answer (yes), but also a sentence summarizing your evidence P for it. You know that your partner will be drawing conclusions from your contribution as you are making it. Accordingly, you'd like to abbreviate your response if the answer is clear from the evidence—to avoid wasting your partner's time.

To test whether you can answer indirectly, you consider the state of the discourse as represented by the content of $[\mathrm{CP}]$. Suppose further that you are in a state in which you have communicated the content of your evidence, so that $[\mathrm{CP}]P$ is true. In that hypothetical context, you want to know whether your partner infers Q. Formalized in our logic of action, these operations correspond to the formula

$$[\mathrm{CP}]([\mathrm{CP}]P \supset [\mathrm{A}]Q)$$

Proving this formula as a query would justify the use of an indirect answer.

The general problem of sentence planning requires more complicated queries along the same lines. For example, the SPUD sentence planner (Stone and Doran, 1997) uses a lexicalized grammar to add content word-by-word to a sentence until a set of communicative goals are satisfied. To test its progress, it uses the similar query

$$[\mathrm{CP}]([\mathrm{CP}]W \supset [\mathrm{A}]G)$$

where W represents the conjunction of the meanings of the words added thus far to the sentence, and G represents the facts SPUD has been asked to see communicated. As with the previous example, making these queries allows SPUD to abbreviate its contributions to conversation and avoid redundancy. To illustrate this, we adapt an observation due to (McDonald, 1992). In the context established by (1a), the bracketed adjuncts in (1b) will be inferrable (given some basic knowledge about presidents and management changeover):

(1) a Nixon resigned from the presidency in 1974.
 b Ford succeeded him [as president in 1974].

SPUD can recognize that these inferences are implicitly available as part of the discourse, and will therefore choose not to include the adjuncts.

As a final example, consider discourse planning for multi-sentential texts. With multiple sentences, we must explicitly represent the update to the discourse made by each sentence and the requirements on the discourse for the sentence to be felicitous. Using a theory of knowledge and communication, we can summarize this information in a clause of the following form—for any sentence u, uttered normally (*done*), with presupposition p and content c:

$$[\mathrm{CP}]([\mathrm{CP}]p \wedge done(u) \supset [\mathrm{CP}]c)$$

Again using implication to represent change, we define a relation $can(1)$ to check whether we can communicate some desired facts G in one step:

$$can(1) \equiv \exists u[\mathrm{CP}](done(u) \supset [\mathrm{CP}]G)$$

But now, following for example (Davis, 1994), we generalize to multiple steps as follows:

$$can(n+1) \equiv \exists u[\mathrm{CP}](done(u) \supset [\mathrm{CP}]can(n))$$

That is, you can achieve a goal in $n+1$ steps if you know what you can do next so that afterwards you can go on to

[1] I also adopt the so-called INCREASING DOMAIN CONSTRAINT, so that for example $[\mathrm{S}]\forall x.p(x) \supset \forall x.[\mathrm{S}]p(x)$, but not necessarily vice versa. Note that when x represents a future situation, this constraint encodes that an agent always maintains knowledge about the future, but may learn more.

achieve the goal in n steps; knowing what to do is expressed using an existential quantifier. Thus, for *each step* in the plan, an additional level of nested knowledge includes the additional information that is available after that action is taken; the new information may then be used to select what the subsequent action should be.

Now let's look at an example requiring such reasoning. S wishes to find out from a bank teller A the balance in S's account, number 42. S plans to say: "I have account 42. What is the balance in my account?" The first sentence has the presupposition that account 42 is mutually identifiable: that there is an a satisfying $\varphi_0(a)$ as defined below:

$$\varphi_0(a) \equiv \quad [\mathrm{CP}]\ (\mathit{account}(a) \wedge \mathit{number}(a, 42) \wedge \mathit{identifiable}(a))$$

Once the first sentence succeeds, the second requires that the balance b in S's account a be mutually identifiable, and that the teller know its value, formalized as $\varphi_1(a, b)$:

$$\begin{aligned}\varphi_1(a, b) \equiv \quad & [\mathrm{CP}]\ (\mathit{account}(a) \wedge \mathit{belongs}(a, S) \wedge \\ & \quad \mathit{balance}(a, b) \wedge \mathit{identifiable}(b)) \wedge \\ & \exists v.\ [\mathrm{A}]\ (\mathit{value}(b, v))\end{aligned}$$

We can describe the preconditions and effects of utterances in rules:

$$[\mathrm{CP}]\forall a(\varphi_0(a) \wedge \mathit{done}(u_0(a)) \supset [\mathrm{CP}]\mathit{belongs}(a, S))$$
$$[\mathrm{CP}]\forall ab(\varphi_1(a, b) \wedge \mathit{done}(u_1(a, b)) \supset [\mathrm{CP}]\mathit{answerable})$$

What we would then like to do is prove $\mathit{can}(2)$ with $G = \mathit{answerable}$; this could not only check that S's contribution to discourse is sensible, it could use unification to arrive at that contribution. By assuming the occurrences of events rather than unifying, we could get an abductive discourse planner (cf. Thomason and Hobbs's contribution).

However, to prove this query, we need some basic facts about banking. In fact, the whole point of formulating the query is to allow us to draw on a variety of shared knowledge about banks and banking in streamlining the discourse. To facilitate reuse, we can specify this information with a special modality $[\mathrm{BANK}]$; we can add another modality $[\mathrm{TELLER}]$ for what any teller knows. Since A is the teller now, this is subject to $[\mathrm{TELLER}]P \supset [\mathrm{A}]P$.

People familiar with banks know that accounts are named by codes like 42:

$$\begin{aligned}1 \quad & [\mathrm{BANK}]\ \forall c.(\mathit{code}(c) \supset \\ & \quad \exists a\ [\mathrm{BANK}]\ (\mathit{account}(a) \wedge \mathit{number}(a, c) \wedge \\ & \quad\quad \mathit{identifiable}(a))) \\ 2 \quad & [\mathrm{BANK}]\ \mathit{code}(42)\end{aligned}$$

They also know that there is a balance for any account, that it is identifiable if the account is, that the teller knows its value:

$$\begin{aligned}3 \quad & [\mathrm{BANK}]\ \forall a\ \exists b\ [\mathrm{BANK}]\ (\mathit{account}(a) \supset \mathit{balance}(a, b)) \\ 4 \quad & [\mathrm{BANK}]\ \forall ab\ (\mathit{balance}(a, b) \wedge \mathit{identifiable}(a) \supset \\ & \quad \mathit{identifiable}(b)) \\ 5 \quad & [\mathrm{BANK}]\ \forall ab(\mathit{balance}(a, b) \supset \exists v.\ [\mathrm{TELLER}]\mathit{value}(b, v))\end{aligned}$$

By adding these facts, it becomes possible to prove the query. The strength of this inferential model is that the presuppositions and other facts that this discourse relies on can be explicitly represented without being explicitly communicated. In a robust system, such facts cannot be ignored altogether; for instance, they are needed to answer questions of clarification (Moore and Paris, 1993). However, they cannot be uttered either—imagine: "I have an account. It's number 42. My account has a balance. What is it?". Such distracting restatements of the obvious have plagued earlier conversational agents, mine included (Power, 1977; Houghton, 1986; Cassell *et al.*, 1994).

We have seen how, in collaborative dialogue, the speaker must model the evolution of private and shared knowledge as information accrues in a discourse. Logic can help—provided it delivers conclusions in a timely manner. The kinds of queries used above are among the most challenging to execute, because nested implications and existential quantifiers require complex proof-theoretic representations. My ongoing research addresses these difficulties by proposing efficient constraint algorithms to manage term representations of possible worlds (Stone, 1997). The culmination of this research is a new, fast modal logic programming language, DIALUP; DIALUP uses constraints to interpret modal logic specifications such as those illustrated above with guaranteed, practical performance. DIALUP is implemented and is already integrated with the SPUD sentence planner.

References

Justine Cassell, Matthew Stone, Brett Douville, Scott Prevost, Brett Achorn, Mark Steedman, Norm Badler, and Catherine Pelachaud. Modeling the interaction between speech and gesture. In *Proceedings of the Cognitive Science Society*, 1994.

Herbert H. Clark. *Using Language*. Cambridge University Press, Cambridge, UK, 1996.

Ernest Davis. Knowledge preconditions for plans. *Journal of Logic and Computation*, 4(5):721–766, 1994.

Nancy Green and Sandra Carberry. A hybrid reasoning model for indirect answers. In *Proceedings of ACL*, pages 58–65, 1994.

George Houghton. *The Production of Language in Dialogue: A Computational Model*. PhD thesis, University of Sussex, 1986.

David McDonald. Type-driven suppression of redundancy in the generation of inference-rich reports. In *Proceedings of INLG 6*, LNAI 587, pages 73–88. Springer, Berlin, 1992.

Johanna D. Moore and Cécile L. Paris. Planning text for advisory dialogues: capturing intentional and rhetorical information. *Computational Linguistics*, 19(4):651–695, 1993.

Robert C. Moore. A formal theory of knowledge and action. In *Formal Theories of the Commonsense World*, pages 319–358. Ablex, Norwood NJ, 1985.

Leora Morgenstern. Knowledge preconditions for actions and plans. In *Proceedings of IJCAI*, pages 867–874, 1987.

Richard Power. The organisation of purposeful dialogues. *Linguistics*, 17:107–152, 1977.

Matthew Stone and Christine Doran. Sentence planning as description using tree-adjoining grammar. In *Proceedings of ACL*, pages 198–205, 1997.

Matthew Stone. Efficient constraints on possible worlds for reasoning about necessity. Technical Report IRCS 97-7 and CIS 97-10, University of Pennsylvania, 1997.