Dialogue Understanding in a Logic of Action and Belief

Alfredo Gabaldon* and **Pat Langley**
Silicon Valley Campus
Carnegie Mellon University
Moffett Field, CA 94035 USA
alfredo.gabaldon@sv.cmu.edu
patrick.w.langley@gmail.com

Abstract

In recent work, Langley et al. (2014) introduced UMBRA, a system for plan and dialogue understanding. The program applies a form of abductive inference to generate explanations incrementally from relational descriptions of observed behavior and knowledge in the form of rules. Although UMBRA's creators described the system architecture, knowledge, and inferences, along with experimental studies of its operation, they did not provide a formalization of its structures or processes. In this paper, we analyze both aspects of the architecture in terms of the Situation Calculus—a classical logic for reasoning about dynamical systems—and give a specification of the inference task the system performs. After this, we state some properties of this formalization that are desirable for the task of incremental dialogue understanding. We conclude by discussing related work and describing our plans for additional research.

Introduction

Humans and other agents who engage in dialogue with others must build a model of other agents' mental states and update their own beliefs and goals as the dialogue progresses. This is particularly important when the purpose of communication is the joint execution of a task. We refer to this mental ascription problem as *dialogue understanding*. This problem is particularly challenging for agents who engage in realistic dialogues, where much is left unsaid by participants who assume substantial background knowledge and ability to infer implicit content.

In this paper, we present a formalization of dialogue understanding in the situation calculus (McCarthy and Hayes 1969)—a logical language for representing and reasoning about dynamic domains—extended with belief and goal modalities. The analysis is inspired by UMBRA, a system for plan and dialogue understanding (Langley et al. 2014; Meadows, Langley, and Emery 2013), and it adopts the same types of knowledge. Our formalization is novel and of independent interest, but it also contributes by providing a formal representation of UMBRA's knowledge structures and a specification of the dialogue understanding problem it addresses. Our specification is only an approximate description, in that we do not attempt to capture why UMBRA favors one explanation over another. The next section provides a brief overview of this system.

After we present our formalization of dialogue understanding, we show three types of desirable properties that follow from it. First, we demonstrate that the formal framework correctly incorporates UMBRA's knowledge structures. Second, we clarify a relationship between two different specifications of dialogue understanding. Finally, we reveal a property of dialogues and their explanations in terms of *common ground* (Clark 1996). We close with a discussion of related research and plans for future work.

A Dialogue Understanding Architecture

Langley et al. (2014) describe UMBRA, a system that incrementally processes observed activities, including dialogues, by computing explanations using a form of abduction. One of our goals is to develop a logical formalization of the knowledge used in UMBRA and the understanding problem it addresses. We make some simplifying assumptions, but we believe our specification is close enough to be valuable and can be extended to capture more of UMBRA's current functionality. Moreover, future development of UMBRA may proceed alongside the formal specification.

UMBRA incorporates three key theoretical assumptions: (a) that dialogue understanding relies on inference about the mental states of participants, (b) that it is an inherently abductive task, and (c) that it proceeds incrementally, which is both realistic and computationally tractable.

The system assumes a *working memory* that stores the mental states of dialogue participants at a given moment. These states are described in terms of *beliefs* and *goals* about the physical world and about other agents. UMBRA uses a declarative representation similar to predicate logic. Examples of working memory elements are *belief(medic, has-injury(p1, i1))* and *goal(expert, stable(p1))*. Beliefs and goals can be nested and have two temporal arguments that encode an interval during which a given literal holds. Additional elements express linear constraints on time points. We omit these temporal arguments, constraints, and the reasoning needed to enforce them. Our logic uses a form of temporal ordering on events, but not one based on time points.

*Current affiliation: GE Global Research, One Research Circle, Niskayuna, NY 12309

The content of nonnested beliefs and goals is literals, such as *has-injury(p1)*, that represent domain information. Since the aim is to understand dialogues, UMBRA also incorporates a notation for six types for speech acts (Austin 1962; Searle 1969): inform, acknowledge, question, propose, accept, reject. The system also expresses these in a predicate-like notation, such as *inform(speaker, listener, content)*, that it can embed in beliefs within working memory.

UMBRA also incorporates a *knowledge* store – essentially a set of rules that encode different types of domain-specific and domain-independent knowledge. Unlike the contents of working memory, such knowledge is stable and changes rarely. The system utilizes four types of knowledge:

- *Speech act rules*, one for each speech act type, that describe the conditions and effects of a speech act on the mental state of agents in a domain-independent manner.
- *Conceptual rules* that define complex predicates in terms of simpler ones, forming a concept hierarchy.
- *Goal-generating rules* that encode conditions which give rise to new goals.
- *Dialogue grammar rules* that specify the patterns of speech acts that constitute a well-formed dialogue.

We will describe these rules in more detail and provide examples when we formalize them later in the paper.

In terms of computational mechanisms, UMBRA processes dialogues in an on-line manner, receiving beliefs about the occurrence of speech acts and attempting to explain them as they arrive. An explanation takes the form of a directed graph, similar to a proof tree, with the nodes being elements in working memory. UMBRA builds such an explanation incrementally, applying a form of abductive inference that extends the explanation, making default assumptions as it attempts to show the observed speech acts combine into a well-formed dialogue.

Formal Preliminaries

In this section, we give an overview of the logical framework on which we base our formalization: the situation calculus (McCarthy and Hayes 1969). In particular, we use an axiomatization that includes Reiter's (1991) solution to the Frame Problem and its extension with modalities for *knowledge* (Scherl and Levesque 2003) and *goals* (Shapiro, Lesperance, and Levesque 1998; 2005).

The situation calculus is a predicate logic language for representing dynamic domains. Its ontology includes *actions* that cause the change, *situations* that represent possible states of the domain, and domain *objects* that include a set of agents. The (actual) initial situation of the domain is encoded by the constant S_0, and the set of situations an agent may believe possible is denoted by the predicate *Init(s)*. The situation that results from executing an action a in a situation s is specified by the function $do(a, s)$. A set of domain-independent axioms ensures that $do(a, s)$ is a unique situation, creating tree-like structures for situations with initial situations as the roots. By nesting the function do, one can create sequences of actions $do(a_n, do(a_{n-1}, \ldots, do(a_1, s) \ldots))$. We will use the notation $do([a_1, \ldots, a_n], s)$ as shorthand for such a situation.

A relation $s \sqsubseteq s'$ represents the existence of a path from s to s'. The domain properties that change when actions are executed are called *fluents* and are represented by predicates with a situation argument. For example, *critical*(p, s) may specify that a patient p is in critical condition in s. The effect of actions on fluents is defined by a set of *successor state axioms*, which capture succinctly the effect of actions on fluents and incorporate a solution to the frame problem (Reiter 1991). We will not be concerned here with actions that affect the physical world, so the only successor state axioms we show are those for two special fluents, $B(ag, s'.s)$ and $G(ag, s', s)$. We use these to model a possible-world semantics for beliefs and goals in the situation calculus. This approach to beliefs and goals is based on the work of Moore (1985) and the extensions of this work cited above.

The fluent $B(ag, s', s)$ means intuitively that, in situation s, agent ag believes s' may be the actual situation.[1] Similarly, $G(ag, s', s)$ states that, in situation s, ag considers s' to be consistent with what it wants to achieve. In order to make B and G behave as intended, one includes a number of axioms that constrain these relations to be transitive and Euclidean, and that all situations B-related to an initial situation are also initial. We defer to the cited work for details.

We can then define the belief and goal modalities *belief*(ag, ϕ, s) and *goal*(ag, ϕ, s) as macros in terms of B and G:

- An agent believes formula ϕ holds in a situation s if it holds in all B-accessible situations s':

$$\textit{belief}(ag, \phi, s) \stackrel{\text{def}}{=} (\forall s')(B(ag, s', s) \supset \phi[s'])$$

- An agent has goal ϕ in situation s if ϕ holds in all situations s' that are G-accessible and there is a path from a current possible (B-accessible) situation to s':[2]

$$\textit{goal}(ag, \phi, s) \stackrel{\text{def}}{=} (\forall s', s'')(G(ag, s', s) \wedge B(ag, s'', s) \wedge s'' \sqsubseteq s' \supset \phi[s'])$$

Note that these definitions use formulae ϕ in reified form. We also use the notation $\phi[s]$, where ϕ denotes a formula in which the situation argument in fluents is replaced by a placeholder. Then $\phi[s]$ denotes the formula obtained by replacing variable s for the placeholder. We use this notation with macros as well. For more details on this encoding of formulae, we refer the reader to (De Giacomo, Lesperance, and Levesque 2000).

Finally, we use a reserved predicate $Poss(a, s)$ and a set of *action precondition axioms* to specify the conditions that make it possible to execute an action in a given situation. Later we will use these axioms to formalize the preconditions of speech acts.

[1] Scherl and Levesque (2003; 2003) and Scherl, Lesperance, and Levesque (1998; 1998; 2005) instead use the predicate $K(ag, s', s)$ and focus on knowledge rather than belief.

[2] The definition used by Shapiro et al. differs slightly by allowing maintenance goals in addition to achievement goals.

Representational Assumptions

We assume a situation calculus language that includes predicate and function symbols to represent the domain of interest. We also include action functions to represent speech acts in an almost identical form as UMBRA, such as $inform(ag_s, ag_l, \ell)$. However, recall that these are action terms and ℓ is a situation calculus formula with situation variables replaced by a placeholder.

The initial mental state of the agents is given by a set of ground facts

$$belief(Ag, L, S_0)$$

that correspond to the initial contents of working memory before any speech acts occur. As in UMBRA, we assume that L is a ground *literal* (an atomic formula or its negation).

Speech Act Rules

UMBRA specifies the preconditions and effects of a speech act in a speech act rule. Consider the simplified rule for the *inform* speech act:

$$\begin{aligned}
&inform(S, L, C, T_1) \leftarrow \\
&\quad belief(S, C, T_1), \\
&\quad goal(S, belief(L, C, T_2), T_1), \\
&\quad belief(S, belief(L, C, T_2), T_2), \\
&\quad belief(L, goal(S, belief(L, C, T_2), T_2), T_2), \\
&\quad belief(L, belief(S, C, T_1), T_2), \\
&\quad belief(L, C, T_2), \\
&\quad T_1 < T_2.
\end{aligned}$$

UMBRA uses a temporal argument to distinguish preconditions from effects. The first two literals in the body, with temporal argument T_1, of this rule are the conditions. The other literals, except for the temporal constraint, are effects.

From the preconditions specified by the inform speech act, we obtain a precondition axiom for inform:[3]

$$\begin{aligned}
Poss(inform(a_s, a_l, \ell), s) \equiv& \\
&belief(a_s, \ell, s) \wedge \\
&goal(a_s, belief(a_l, \ell), s) \ .
\end{aligned}$$

As mentioned earlier, speech acts only affect the B and G accessibility relations, so their effects are captured in the successor state axioms for these relations. The successor state axiom for B has the form

$$\begin{aligned}
&B(ag, s'', do(a, s)) \equiv \\
&(\exists s')[B(ag, s', s) \wedge s'' = do(a, s') \wedge Poss(a, s') \wedge \\
&(\forall ag_s, ag_l, \ell)(a = inform(ag_s, ag_l, \ell) \supset \Phi_{inf} \wedge \\
&\quad a = acknowledge(ag_s, ag_l, \ell) \supset \Phi_{ack} \wedge \\
&\quad a = \ldots)]
\end{aligned} \tag{1}$$

where Φ_{inf} stands for

$$\begin{aligned}
&ag = ag_s \supset belief(ag_l, \ell, s') \wedge \\
&ag = ag_l \supset (goal(ag_s, belief(a_l, \ell), s') \ \wedge \ \ell[s']) \wedge \\
&ag \neq ag_s \wedge ag \neq ag_l \supset \\
&\qquad \{belief(ag_s, belief(ag_l, \ell), s'') \wedge \\
&\qquad\quad belief(ag_l, goal(ag_s, belief(ag_l, \ell)), s'') \wedge \\
&\qquad\quad belief(ag_l, belief(ag_s, \ell), s') \wedge \\
&\qquad\quad belief(ag_l, \ell, s')\}
\end{aligned}$$

[3] Throughout the paper, we assume variables that appear free in formulae are implicitly universally quantified with widest scope.

and Φ_{sp} for other speech acts sp are similar in their structure.

The successor state axiom for G is analogous but simpler, since the only speech act that affects it directly is *accept*:

$$\begin{aligned}
&G(ag, s', do(a, s)) \equiv \\
&G(ag, s', s) \wedge (\exists s'')(B(ag, s'', s) \wedge do(a, s'') \sqsubseteq s') \wedge \\
&\quad (\forall ag_l, \ell)(a = accept(ag, ag_l, \ell) \supset \ell[s'])].
\end{aligned} \tag{2}$$

This axiom accounts for achievement goals and the one speech act, *accept*, that has an effect on relation G. As Shapiro et al. (2005) have shown, it can be generalized to handle *maintenance goals* (i.e., goals for something to hold over a period of time) and to handle events that cancel an existing goal, such as a *cancel_propose* speech act.

Conceptual Rules

Consider now the knowledge that UMBRA encodes in its conceptual rules, which define complex concepts in terms of simpler ones. As part of a theory of actions, these rules can be cast as *state constraints* on situations, which are also called *static laws*. Dealing with state constraints is a challenging task known in AI as the *ramification problem*, especially when the constraints include cycles. Fortunately, due to the hierarchical nature of the concept definitions, conceptual rules are acyclic and therefore simpler to handle.

However, before we can incorporate concept definitions, we must first express the conceptual rules' knowledge in our logical language. A simple example of such a rule is: "A patient is in critical condition if she has an artery bleeding," which we may encode as the implication

$$\begin{aligned}
(body_part(p, b, s) \wedge artery(b, s) \wedge bleeding(b, s)) \\
\supset critical(p, s) \ .
\end{aligned}$$

The fact that all literals in this formula have the same situation argument s indicates that it is indeed a state constraint. In general, conceptual rules can be encoded as formulae of the form

$$F(\vec{y}, s) \supset p(\vec{x}, s) \ , \tag{3}$$

where the only free variables in formula $F(\vec{y}, s)$ are $\vec{y}$ and s, the only situation term that appears in it is s, and p is a fluent predicate.

If an agent is informed that $critical(P_1)$, UMBRA's explanation mechanism would infer, assuming the above rule is the only one with *critical* in the head, that P_1 is bleeding from an artery. In other words, the system makes an implicit assumption that, for a concept p, the set of all conceptual rules with head p constitute a complete definition of that concept. In order to capture this behavior in our formalization, we must gather, for each high-level concept, all the conceptual rules that define it and put their encoding into the form of an equivalence,

$$\mathbf{F}(\vec{y}, s) \equiv p(\vec{x}, s) \ , \tag{4}$$

where $\mathbf{F}(\vec{y}, s)$ stands for the formula $F_1(\vec{y}_1, s) \vee \ldots \vee F_n(\vec{y}_n, s)$ such that each disjunct corresponds to the antecedent of a conceptual rule for p.

Assuming that all agents utilize their conceptual rules, incorporating them into the formalization means that all the worlds considered possible by an agent should satisfy the rules. Thus we must add them as conditions that are satisfied by situation s'' in the successor state axiom (1) for B. For each concept p defined by the conceptual rules, we would add the conjunct

$$\mathbf{F}(\vec{y}, s') \equiv p(\vec{x}, s') \ . \tag{5}$$

For the example rule above, we would add the condition

$$(\mathit{body\_part}(p, b, s') \wedge \ldots) \equiv \mathit{critical}(p, s') \ .$$

This means that, if an agent is informed that $\mathit{critical}(P_1)$, not only will this hold in all the resulting situations believed possible, but so will $\mathit{body\_part}(P_1, B_1)$, $\mathit{artery}(B_1)$, and $\mathit{bleeding}(B_1)$ for some object B_1. In other words, the agent will come to believe these literals.

Goal-Generating Rules

Another form of knowledge describes situations in which new goals arise. UMBRA encodes this content in terms of *goal-generating rules*, which are similar to conceptual rules except that their heads are literals that become goals when the rule applies. If a rule's body is satisfied by the current contents of working memory, then the system adds the instantiated head to memory as a new goal. An example is a rule that states: "If an agent believes that a patient is in critical condition, then it has the goal of getting the patient in stable condition."

We can express goal-generating rules in our logic as formulae of the form

$$\begin{array}{r} \mathit{belief}(ag, f_1(\vec{y}_1), s) \wedge \ldots \wedge \mathit{belief}(ag, f_n(\vec{y}_n), s) \\ \supset \mathit{goal}(ag, p(\vec{x}), s) \end{array} \tag{6}$$

with one example being

$$\mathit{belief}(ag, \mathit{critical}(p), s) \supset \mathit{goal}(ag, \mathit{stable}(p), s) \ .$$

As with conceptual rules, the presence of formulae like (6) in our formalization means that we have another instance of the ramification problem, but this time for goals. Like conceptual rules, these are stratified, but this involves only two strata, since the antecedents are expressed in terms of belief, while the consequents are instead stated in terms of the goal modality.

We can then incorporate formulae (6) from goal generating rules as additional conditions in the successor state axiom (2) of relation G. We add these conditions as conjuncts of the form:

$$\bigwedge_{i=1,\ldots,n} \mathit{belief}(ag, f_i(\vec{y}_i), do(a, s)) \supset p(\vec{x}, s') \ .$$

Dialogue Grammar Rules

We have described our formalization of the direct and indirect effects of speech acts on agents' mental states as encoded by the B and G relations. We need one last form of knowledge before we can specify the dialogue understanding problem, namely, knowledge about what counts as a well-formed dialogue. UMBRA stores this knowledge in a "dialogue grammar" that it also encodes as rules. Intuitively, this grammar specifies what sequences of speech acts are acceptable dialogues. The grammar imposes conditions on the types, the agents, and the content of the speech acts that may appear in a legitimate conversation.

In our analysis, we will use formulae with a similar but slightly simpler grammar-like structure. We will define a dialogue as a sequence of *adjacency pairs* (Sacks, Schegloff, and Jefferson 1974) and generalizations of this idea: *inform-acknowledge*, *question-inform-acknowledge*, *propose-accept* and *propose-reject-inform-acknowledge*. We refer to these short sequences of speech acts as *exchanges* and define them by means of macros:

- An *inform-acknowledge* exchange is a corresponding pair of speech acts with matching speaker, listener and content arguments:

$$\begin{array}{l} \mathit{inform\_ack}(a_1, a_2, s) \stackrel{\text{def}}{=} (\exists ag_s, ag_l, \ell) \\ \qquad a_1 = \mathit{inform}(ag_s, ag_l, \ell) \wedge \\ \qquad a_2 = \mathit{acknowledge}(ag_l, ag_s, \ell). \end{array}$$

- The *question-inform-acknowledge* exchange, in addition to the right pattern of arguments, requires that the content of the inform act be relevant to the content of the question. We do not define *relevant* here, but its intended purpose is to capture whether the content of the inform following a question can be considered an answer.

$$\begin{array}{l} \mathit{question\_inform\_ack}(a_1, a_2, a_3, s) \stackrel{\text{def}}{=} (\exists ag_s, ag_l, \ell, \ell') \\ \qquad a_1 = \mathit{question}(ag_s, ag_l, \ell) \wedge \\ \qquad a_2 = \mathit{inform}(ag_l, ag_s, \ell') \wedge \\ \qquad a_3 = \mathit{acknowledge}(ag_s, ag_l, \ell') \wedge \\ \qquad relevant(\ell, \ell', s). \end{array}$$

  The *propose-accept* exchange is similar to *inform-acknowledge* and we omit it.

- In a *propose-reject-inform-ack* exchange, an agent rejects a proposal and informs the other agent of its reason for rejecting it:

$$\begin{array}{l} \mathit{propose\_reject\_inform\_ack}(a_1, a_2, a_3, a_4, s) \stackrel{\text{def}}{=} \\ (\exists ag_s, ag_l, \ell, \ell') a_1 = \mathit{propose}(ag_s, ag_l, \ell) \wedge \\ \qquad a_2 = \mathit{reject}(ag_l, ag_s, \ell) \wedge \\ \qquad a_3 = \mathit{inform}(ag_l, ag_s, \ell') \wedge \\ \qquad a_4 = \mathit{acknowledge}(ag_s, ag_l, \ell') \wedge \\ \qquad relevant(\ell, \ell', s). \end{array}$$

In addition, we introduce a macro for situations that correspond to a sequence of actions that ends in one of the above exchanges. Intuitively, $\mathit{exchange}(s', s)$ means that executing one of the exchanges in situation s' results in situation s:

$$\begin{array}{l} \mathit{exchange}(s', s) \stackrel{\text{def}}{=} (\exists a_1, a_2, a_3, a_4) \\ s = do(a_1, do(a_2, s')) \wedge \mathit{inform\_ack}(a_2, a_1, s') \vee \\ s = do(a_1, do(a_2, do(a_3, s'))) \wedge \\ \qquad \mathit{question\_inform\_ack}(a_3, a_2, a_1, s') \vee \\ s = do(a_1, do(a_2, s')) \wedge \mathit{propose\_accept}(a_2, a_1, s') \vee \\ s = do(a_1, do(a_2, do(a_3, do(a_4, s')))) \wedge \\ \qquad \mathit{propose\_reject\_inform\_ack}(a_4, a_3, a_2, a_1, s'). \end{array}$$

Finally, we define $\mathit{dialogue}(s)$ to denote a (possibly empty) dialogue has led to situation s. This holds if s is an initial situation or a dialogue followed by an exchange leads to s:

$$\mathit{dialogue}(s) \equiv s = S_0 \vee (\exists s') \mathit{exchange}(s', s) \wedge \mathit{dialogue}(s').$$

Dialogue Understanding

We can now state the problem of *dialogue understanding*. Given a sequence of observed speech acts, the task is to find (a) a set of belief and goal facts and (b) implicit speech acts such that a background theory together with the facts entails that the sequence of speech acts is a well-formed dialogue.

We will consider two versions of this task. The first is a *global* dialogue understanding problem in which the observed speech acts are given at once. The second task is an *incremental* version in which the observations are given sequentially and the explanation is obtained in an iterative fashion. The latter comes closer to the way that humans and UMBRA interpret dialogues.

To continue, we must introduce a few concepts and notation. *Belief facts* and *goal facts* are ground atomic formulae of the form *belief*(Ag, ℓ, S), respectively, *goal*(Ag, ℓ, S). By *fact* we mean a belief or a goal fact. We say a fact is *relative to situation* S if S is its situation argument. Let T be a theory comprised of four sets:

- T_{sc}, the foundational situation calculus axioms;
- T_{sa}, a speech act theory including the axioms for modalities B and G;
- T_g, the dialogue grammar axioms; and
- T_{S_0}, the initial mental state.

We say a *dialogue* is a situation term $do([Sp_1, \ldots, Sp_n], S_0)$ where each Sp_i is a ground speech act term. Let $\widehat{D}$, which is called an *extension* of $D = do([Sp_1, \ldots, Sp_n], S_0)$, be a term of the form $do([Sp'_1, \ldots, Sp'_m], S_0)$, such that $n \leq m$ and there is an injective function $f : \{1, \ldots, n\} \mapsto \{1, \ldots, m\}$, $f(i) < f(j)$ if $i < j$, and $Sp_i = Sp'_{f(i)}$.

In other words, all speech acts of D appear in $\widehat{D}$ with their order preserved. We also need a macro *executable*(s), which means that each action in a sequence is possible:

$$executable(s) \stackrel{\text{def}}{=} (\forall a, s').do(a, s') \sqsubseteq s \supset Poss(a, s')$$

and the notion of a *global hypothesis for dialogue* D, which is a set $GH_{\widehat{D}}$ of facts each relative to a situation subterm of $do(\widehat{D}, S_0)$, where $\widehat{D}$ is equal to or an extension of D.

Definition 1 (Global Explanation) *The global hypothesis* $GH_{\widehat{D}}$ *of* D *is a* global explanation *of a dialogue* D *if*

- $T \cup GH_{\widehat{D}}$ *is consistent and*
- $T \cup GH_{\widehat{D}} \models$ *executable*$(do(\widehat{D}, S_0)) \wedge dialogue(do(\widehat{D}, S_0))$.

It is clear that this problem involves a form of abductive inference. Indeed, the above definition is rather close in form to the standard definition of logic-based abduction (e.g., see Eiter and Gottlob 1995). One glaring omission is the absence of any preference, such as minimality, on explanations. We leave this aspect underspecified and assume that some criterion is used uniformly.

In the incremental form of the problem, in addition to an observed partial dialogue, we are also given the explanation obtained in the preceding iteration. This leads to the concept of a *local hypothesis* for a dialogue D, which is a set LH_D of facts all relative to $do(D, S_0)$.

Let D' be a dialogue, E' be a set of facts each relative to a subterm of $do(D', S_0)$, and Sp be a ground speech act term. Intuitively, D' and E' are the current dialogue and its explanation prior to the most recent observation Sp.

Definition 2 (Incremental Explanation) *Let* D',E' *and* Sp *be as above and* $D = do(\widehat{Sp}, do(D', S_0))$. *A local hypothesis* LH_D *of dialogue* D *is a* local explanation of Sp with respect to D' and E' *if*

- $T \cup E' \cup LH_D$ *is consistent and*
- $T \cup E' \cup LH_D \models$ *executable*$(do(\widehat{Sp}, do(D', S_0))) \wedge$ *dialogue*$(do(\widehat{Sp}, do(D', S_0)))$.

Notice that, according to this definition, the prior dialogue D' is not expanded as part of the explanation. The prior explanation E' is also preserved, since the incremental nature of the process is intended to make only "local" decisions and assumptions. We only expand the singleton sequence Sp and the hypothesis contains only facts relative to D.

Building dialogue explanations incrementally by making local assumptions makes the computational problem more tractable, but obviously there is a price to pay. The process may fail to find a local explanation even if a global explanation exists. In the next section we show that, if local explanations are found for a complete dialogue, then their union coincides with a global explanation.

Relevant Properties

With our formal framework in place and our specification of the dialogue understanding task, we can now formulate a number of properties. We consider properties about the framework, about the dialogue understanding formulations, and about the dialogues themselves.

In terms of framework properties, we can show that the successor state axioms of the modalities B and G capture the effects of the speech acts as intended. For instance, for *inform*, we have:

Proposition 1 *Let* A, B *be agents,* S *a ground situation term and* ℓ *a formula. The logical consequences of a theory* T *include:*

- *belief*$(A,$ *belief*$(B, \ell), do($*inform*$(A, B, \ell), S))$;
- *belief*$(B,$ *belief*$(A, \ell), do($*inform*$(A, B, \ell), S))$;
- *belief*$(B,$ *goal*$(A,$ *belief*$(B, \ell)), do($*inform*$(A, B, \ell), S))$;
- *belief*$(B, \ell, do($*inform*$(A, B, \ell), S))$.

We can also verify that the framework correctly incorporates knowledge about conceptual inference and goal generation. Suppose that a formula $F(\vec{y}, s) \equiv p(\vec{x}, s)$, corresponding to conceptual rules for predicate p, has been included in a theory T as described above. Then we have:

Proposition 2

$$T \models belief(ag, F(\vec{y}), s) \supset belief(ag, p(\vec{x}), s).$$

For example, if we have *belief*$(A,$ *body_part*$(P_1, B_1), S)$ and *belief*$(A,$ *artery*$(B_1), S)$ hold, then we will also have *belief*$(A,$ *critical*$(P_1), do($*inform*$(B, A,$ *bleeding*$(P_1)), S))$.

A similar result holds regarding goal-generating knowledge. After incorporating a formula (6), we have:

Proposition 3

$$T \models \bigwedge_{i=1,\ldots,n} \mathit{belief}(ag, f_i(\vec{y}_i), s) \supset \mathit{goal}(ag, p(\vec{x}), s).$$

Consider next our two notions of explanation. One question we raised earlier was whether incremental processing of a dialogue will eventually arrive at a global explanation. The next proposition establishes this property.

Proposition 4 *Suppose that a sequence of observed speech acts $Sp_1, \ldots, Sp_n$ results in a sequence of local explanations $E_1, \ldots, E_n$ and an expanded dialogue $\widehat{D}$ of $D = do([Sp_1, \ldots, Sp_n], S_0)$. Then $E_1 \cup \ldots \cup E_n$ is a global explanation of D.*

As mentioned earlier, even if the converse of this proposition holds and a global explanation exists, incremental dialogue processing may fail to find a sequence of local explanations for the complete conversation. One alternative would be to incorporate backtracking into the explanation process. However, the computational advantage of incremental processing would then be largely lost. We will explore other options in future work.

Finally, an interesting property of dialogues and their explanations involves the important notion of *common ground* (Clark 1996). For a pair of agents engaged in a dialogue, we can define the common ground as the set of literals ℓ mutually believed by both agents in a given situation:[4]

$$\begin{array}{l} cg(ag_1, ag_2, \ell, s) \stackrel{\text{def}}{=} \\ \mathit{belief}(ag_1, \ell, s) \wedge \mathit{belief}(ag_2, \ell, s) \wedge \\ \mathit{belief}(ag_1, \mathit{belief}(ag_2, \ell), s) \wedge \\ \mathit{belief}(ag_2, \mathit{belief}(ag_1, \ell), s) \wedge \\ \mathit{belief}(ag_1, \mathit{belief}(ag_2, \mathit{belief}(ag_1, \ell)), s) \wedge \\ \mathit{belief}(ag_2, \mathit{belief}(ag_1, \mathit{belief}(ag_2, \ell)), s). \end{array}$$

We can then state the property that, as a dialogue progresses and is incrementally explained, the common ground between the agents is nondecreasing. Let D be a dialogue with local explanations $E_1, \ldots, E_n$ and expanded dialogue $\widehat{D}$. Let D_i be the prefix of D that consists of the first i speech acts and $\widehat{D}_i$ be the expanded dialogue that corresponds to local explanation E_i.

Proposition 5 *Let T, $E_1, \ldots, E_n$, D, $\widehat{D}$ be as defined above and $S_k = do(\widehat{D}_k, S_0)$ for any k. If $1 \leq i \leq j \leq n$, then*

$$T \bigcup_{l=1,\ldots,n} E_l \models cg(ag_1, ag_2, \ell, S_i) \supset cg(ag_1, ag_2, \ell, S_j).$$

Related Research

The problem of dialogue understanding has received some attention in the literature. Our formalization builds substantially on earlier work on theories of action, speech act theory, possible worlds approaches to belief, and others analyses.

Our situation calculus framework is based on the work of Scherl and Levesque (2003) and Shapiro, Lesperance, and Levesque (1998; 2005). Our formalization uses their successor state axioms for relations B and G but modifies them to incorporate conceptual and goal-generating knowledge. Other formalisms for belief and goals, based on different logics, include (Herzig and Longin 2000; Martin, Narasamdya, and Thielscher 2004; van Benthem, van Eijck, and Kooi 2006; Baral et al. 2010). Earlier work on speech acts, especially analyses of speech acts' conditions and effects in terms of mental states (Perrault and Allen 1980; Allen and Perrault 1980), has also influenced our formalization in substantial ways.

Above the utterance level, Carberry and Lambert (1999) report a dialogue interpretation system that is also based on speech acts. Their system can recognize subdialogues used by an agent who is convincing another of some proposition, which we have not addressed. Although they described their interpretation mechanism in pseudo-code, rather than declaratively, their "discourse recipes" specify speech acts in terms of agents' beliefs.

We have formalized the dialogue understanding problem in terms of logical abduction, and some earlier work has taken a similar approach. For instance, Litman and Allen (1985; 1987) present an abductive approach to dialogue processing, as does work reported by McRoy and Hirst (1995) that includes mechanisms for recovering from misunderstandings. Hobbs et al. (1993) have also used abductive inference for understanding, but only for processing at the sentence level.

Finally, we should mention important prior research on formalizations of discourse structure, such as that arising in dialogue. This includes the comprehensive work of Asher and Lascarides (2003), which extends Discourse Representation Theory (Kamp and Reyle 1993), and Rhetorical Structure Theory (Mann and Thompson 1987), among other prominent contributions.

Conclusions

In this paper, we have presented a formalization of dialogue understanding in an extended situation calculus that includes modalities for belief and goals. Our framework builds on speech act theory and encodes both their direct effects on agents' mental states and their indirect effects due to inferences by conceptual and goal-generating knowledge. We provided two specifications of the dialogue interpretation task, and we showed a number of desirable properties about the formalization, the two variants of dialogue understanding, and the evolution of common ground, in terms of shared beliefs and goals, as a conversation progresses.

We believe that our framework for dialogue understanding is valuable both in its own right and as a formal specification for UMBRA, a system that interprets high-level dialogues through a process of incremental abduction. In future research, we intend to elaborate the framework to support a broader set of speech acts, additional forms of belief and goal revision, and forms of limited belief. We also hope to analyze more subtle aspects of conversation, such as whether a speaker's utterances are considerate of the listener's mental state and general sensibilities.

[4] Clearly, what we really need here is a mutual belief modality, but we can still state a useful property without it.

Acknowledgments

This research was supported in part by Grant N00014-09-1-1029 from the Office of Naval Research. We thank Paul Bello, Will Bridewell, and Ben Meadows for useful discussions and the reviewers for valuable comments.

References

Allen, J., and Perrault, C. 1980. Analyzing intention in utterances. *Artificial Intelligence* 15:143–178.

Asher, N., and Lascarides, A. 2003. *Logics of conversation*. Cambridge University Press.

Austin, J. L. 1962. *How to do things with words*. Cambridge, MA: Harvard University Press.

Baral, C.; Gelfond, G.; Son, T. C.; and Pontelli, E. 2010. Using answer set programming to model multi-agent scenarios involving agents' knowledge about other's knowledge. In *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems*, 259–266.

Carberry, S., and Lambert, L. 1999. A process model for recognizing communicative acts and modeling negotiation subdialogues. *Computational Linguistics* 25:1–53.

Clark, H. H. 1996. *Using language*. Cambridge, UK: Cambridge University Press.

De Giacomo, G.; Lesperance, Y.; and Levesque, H. 2000. ConGolog, a concurrent programming language based on the situation calculus. *Artificial Intelligence* 121:109–169.

Eiter, T., and Gottlob, G. 1995. The complexity of logic-based abduction. *Journal of the ACM* 42:3–42.

Herzig, A., and Longin, D. 2000. Belief dynamics in cooperative dialogues. *Journal of Semantics* 17:91–115.

Hobbs, J. R.; Stickel, M. E.; Appelt, D. E.; and Martin, P. 1993. Interpretation as abduction. *Artificial Intelligence* 63:69–142.

Kamp, H., and Reyle, U. 1993. *From discourse to logic*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Langley, P.; Meadows, B.; Gabaldon, A.; and Heald, R. 2014. Abductive understanding of dialogues about joint activities. *Interaction Studies*, 15:426–453.

Litman, D. J., and Allen, J. F. 1987. A plan recognition model for subdialogues in conversations. *Cognitive Science* 11:163–200.

Litman, D. 1985. Plan recognition and discourse analysis: An integrated approach for understanding dialogues. PhD Thesis, Technical Report 170, Department of Computer Science, Univ. of Rochester.

Mann, W. C., and Thompson, S. A. 1987. *Rhetorical structure theory: Description and construction of text structures*. Springer.

Martin, Y.; Narasamdya, I.; and Thielscher, M. 2004. Knowledge of other agents and communicative actions in the fluent calculus. In *Proceedings of the Ninth International Conference on Principles of Knowledge Representation and Reasoning*, 623–633.

McCarthy, J., and Hayes, P. 1969. Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B., and Michie, D., eds., *Machine intelligence 4*. Edinburgh University Press. 463–502.

McRoy, S., and Hirst, G. 1995. The repair of speech act misunderstandings by abductive inference. *Computational Linguistics* 21:435–478.

Meadows, B.; Langley, P.; and Emery, M. 2013. Understanding social interactions using incremental abductive inference. In *Proceedings of the Second Annual Conference on Advances in Cognitive Systems*, 39–56.

Moore, R. C. 1985. A formal theory of knowledge and action. In Hobbs, J. R., and Moore, R. C., eds., *Formal theories of the common sense world*. Norwood, NJ: Ablex Publishing. 319–358.

Perrault, C. R., and Allen, J. F. 1980. A plan-based analysis of indirect speech acts. *Computational Linguistics* 6:167–182.

Reiter, R. 1991. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In Lifschitz, V., ed., *Artificial intelligence and mathematical theory of computation*. New York: Academic Press. 359–380.

Sacks, H.; Schegloff, E. A.; and Jefferson, G. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language* 50:696–735.

Scherl, R. B., and Levesque, H. J. 2003. Knowledge, action, and the frame problem. *Artificial Intelligence* 144:1–39.

Searle, J. 1969. *Speech Acts: An essay in the philosophy of language*. New York: Cambridge University Press.

Shapiro, S.; Lesperance, Y.; and Levesque, H. J. 1998. Specifying communicative multi-agent systems. In *Agents and Multi-Agent Systems Formalisms, Methodologies, and Applications*. Springer. 1–14.

Shapiro, S.; Lesperance, Y.; and Levesque, H. J. 2005. Goal change. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 582–588.

van Benthem, J.; van Eijck, J.; and Kooi, B. 2006. Logics of communication and change. *Information and Computation* 204:1620–1662.