Efficient Dialogue Using a Probabilistic Nested User Model

Bryan McEleney and Gregory O'Hare
Department of Computer Science
University College Dublin
Belfield, Dublin 4
bryan.mceleney@ucd.ie

Abstract

We describe a set of dialogue simulation experiments, in which a probabilistic nested user model is employed in deciding between speech acts for a collaborative planning task, finding that a gain in utility can be obtained by using a probabilistic rather than a logical model. Given a set of ordinary dialogue plan rules, our system generates a game-tree representation of the dialogue, using chance nodes to represent uncertain preconditions in the plan. Then, the game-tree is evaluated with respect to a given user model state.

1 Introduction

It has long been recognised that generation of cooperative dialogue should involve some model of the hearer. In human dialogue, economy of expression depends on prediction of the interpretation that the hearer will make of a dialogue act, within the context of the dialogue history and the beliefs that support the hearer's inferences. Instead of a reflex response, the hearer produces a deep plan structure that fits the dialogue history [Carberry, 1990], and revises his beliefs based on the preconditions, effects, and their consequences associated with each act in the dialogue history. Then, he adds a consistent act to the inferred plan structure. A number of phenomena arise from this model. Appropriate referring expressions can be selected and interpreted [Heeman and Hirst, 1995]. Indirect speech acts can be planned [Allen and Perrault, 1980]. Extra-cooperative behaviour like correction of misconception [McCoy, 1989] and provision of unasked-for information or acts emerges. With different agents planning each act, it is possible that the contributions that each makes to the overall plan structure are founded on different beliefs about the domain state and the applicable plan rules [Pollack, 1986].

While these phenomena are interesting, there is a problem that has received little attention, and that is to employ plan recognition with nested belief models to generate the shortest possible dialogue. Using each agent's ability to recognise plans, the number and execution time of the dialogue acts can be reduced. This is significant problem in online planning in a spoken language dialogue system, as well as in offline planning of dialogue policies. Such policies can be applied to human-human interactions, such as deciding whether to ask the customer "Would you like fries with that?", or to graphical or text-based interface design where communicative acts must be chosen. We argue that the way forward in achieving this is to employ a probabilistic nested belief model, rather than a logical one, and in this paper, we show how much more efficient such dialogues can be once a probabilistic model is adopted.

We are currently applying our planner to the problem of collaborative planning, in which agents negotiate over a plan in which both of them will act. This differs from domain-level planning in which the agents act within the object plan. We have also applied our planner in previous work to pure domain-level planning, in which dialogue acts were specified as domain acts. We have used it to decide between using a clarification subdialogue and risking plan failure, which was a problem suggested by [Carletta, 1992], and have shown that the choice of strategy depends on a probabilistic belief variable. A considerable performance gain was obtained by adapting the strategy to this probabilistic variable. We have used it to decide whether to take the initiative in a task [McEleney and O'Hare, 2005], where the first agent must use a model of a second agent to predict whether the second will take the initiative in beginning a collaborative task. Once again, the fact that a probabilistic model was used made a difference to the efficiency of the dialogues. We have also shown that the planner is capable of working with other well known dialogue problems such as correcting misconceptions, and in planning an indirect speech act. Each problem can be easily specified using a nested belief model.

2 The planner

Our planner rests on an abstract, game-theoretic model of dialogue, in which agents decide between dialogue strategies in pursuit of rewards, by constructing a game-tree representation of the dialogue. Dialogue is planned to the level of the dialogue act, which is a structure with illocutionary force and a propositional content. There is no translation between the dialogue act and a natural language representation. Since we have no useful data of human dialogues within a collaborative planning domain, and no natural language planner, our system uses a fixed cost estimate for each type of dialogue act, which is adequate to characterise dialogue behaviour, if not to obtain concrete results. The planner ignores the lim-

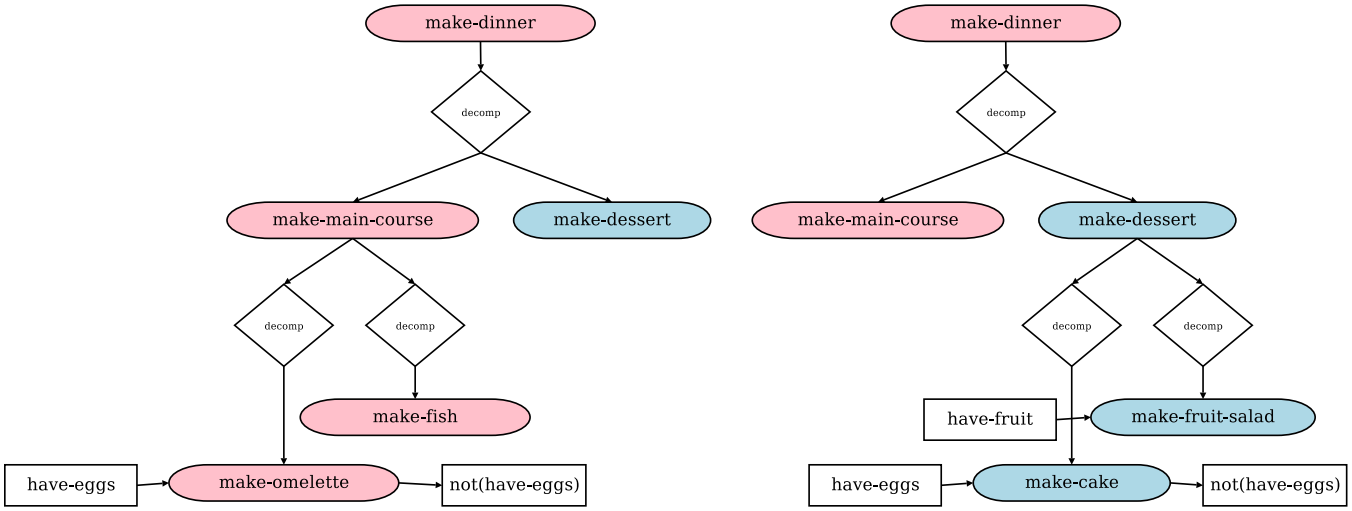

Figure 1: Plan Library

itations of human rationality, and is therefore an unrealistic mechanism for modelling human behaviour, especially as game trees can be large and difficult to construct. Instead we work with perfect utility-maximising players. Nor do we evaluate the planner with system-human trials since a simulation provides much more detailed data about the character of the planning problems that we are investigating. Instead, all the results are for system-system experiments, with a view to later evaluation in a human-system setting.

One important advantage is that our planner is easy to use without expert knowledge of the underlying theory. In fact it is relatively easy to specify a dialogue problem without such knowledge by using the same hierarchical plan rules [Sacerdoti, 1974] that are commonly used to specify dialogue plans. The belief information is then acquired automatically from dialogue histories, so that the system will adapt itself over time. This is done using the planner's belief revision component, which updates beliefs in response to dialogue acts.

Our negotiation model is similar to the RMM of Gmytrasiwicz and Durfee [2001], in which game matrices are used as the nodes in a tree structure that branches on the beliefs of the agents. Our model differs in that we start with a given set of plan rules, from which a game tree is constructed. It is possible to then convert our game tree to normalised form, at which point we would obtain an RMM structure. However, the raw game tree is equally useful. The advantage of our planner is that it can work from the source plan rules, not requiring a game matrix to be provided. The negotiation acts are based on value of information in that an agent computes the change in utility of a plan as a result of obtaining a piece of information. Our set of negotiation acts is similar to those found in this paper.

Input to the planner is a nested belief model. Each level of the belief model has beliefs about the domain state, about the agent's capabilities in terms of hierarchical plan rules, and about plan recognition rules in terms of probability distributions over explanations for a given act in the plan structure. Figure 1 is an illustration of a typical plan library, with the first level of nesting on the left-hand-side, and the second level of nesting on the right-hand-side. In this case we have two libraries since we have two agents with different expertise - one who is good at making main courses, and one who is good at making desserts. This pair is used to initialise levels one and two, levels three and four, and so on. Instead, we could have used only one library, with which to initialise every level, and allowed the system to adapt each level to the statistics in the collection of dialogue histories, by using a belief revision process. For example, if agent two were to often make-cake, the plan rule at level two that make-dessert decomposes to make-cake would attain a high probability of belief, as would the belief have-eggs. In this problem, the first agent must choose between cooking an omelette and cooking fish. The problem is that while the omelette is preferred, it requires eggs, which means that the second agent cannot make a cake for dessert. However, if the second agent has fruit, he can make a fruit salad instead. Therefore the first agent would like to find out whether the second agent has fruit before making his decision, and this is the source of a negotiation dialogue between the agents. This example is perhaps the smallest domain-level plan that can be constructed. In larger problems, agents would construct a deep game tree at of many moves, and this would form the basis of a long negotiation dialogue which deals with many different beliefs.

The planner constructs a domain-level game tree for this problem, illustrated in figure 2. To do this, the first agent chooses an action. Then the first agent performs the plan recognition process that he expects of the second agent. This uses the third level beliefs, since the second agent is working at level two and is trying to use beliefs about the first agent at level three to reconstruct the first act in the plan. In plan recognition, we use a simplifying assumption of "focussing", whereby an agent will always complete one branch of a plan before developing another one. Focussing is something that happens in human dialogue [Grosz and Sidner, 1986] for the very reason that it restricts the plan recognition hypotheses. Under this assumption, the plan recogniser need only search

among focussed candidates to explain the dialogue history. This means that multiple explanations can only occur when the plan tree is full, and a parent must be added to the plan tree to open up a sibling branch. For each act in the agent's repertoire, there is a probability distribution stored for a list of candidate parents. Having performed plan recognition at level three, the plan is expanded by adding an act to the next open node in the plan tree, using the level two plan rules. Suppose a third act were to be added to the plan. Then the agent at level two would expect the agent planning at level three to call the planer at level four to obtain a plan with two acts. In turn, the planner at level four calls the planner at level five. Each of the alternative plan decompositions available to an agent produces an alternative at a choice node in the game tree. If the plan recogniser must add a parent to a full subtree, or if there is a precondition to an act whose satisfaction depends on the belief state of the agent, a chance node is added to the game tree, preceding the choice node.

To illustrate, consider the omelette problem. The first agent decomposes make-dinner to make-main-course and then to either of make-omelette or make-fish, so we end up with two branches in the game tree at the root node (figure 2). Taking the make-fish branch, the second agent is expected to infer the make-main-course parent, and its parent, make-dinner, using the beliefs at level three. These are enough parents to make an open tree, and so make-dessert is attached. make-dessert can be decomposed to make-cake or to make-fruit-salad. Each of these has a precondition, and so a chance node is introduced to the game tree. Then, a choice node is added to each of the chance node outcomes.

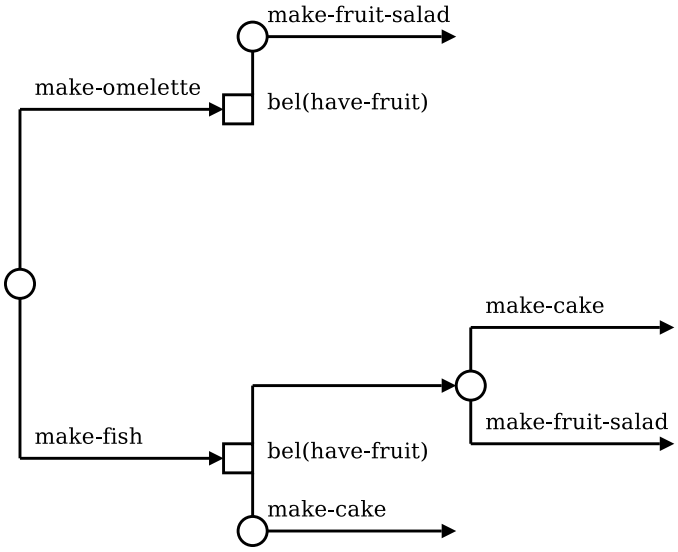

Figure 2: Game Tree

2.1 Evaluation of the game tree

Once constructed, the game tree must be evaluated in the context of a given belief state. Starting at the root of the tree, the first agent's nested belief state is taken, and the probabilities of the chance nodes are found, alternating between level one and level two as the turn passes along the length of the dialogue. The value of a chance node is given as the weighted sum of its children, according to the expected utility rule. To evaluate a choice node, the remainder of the dialogue is evaluated from the point of view of the agent acting at the choice node. For example, the second choice node would be evaluated at level two, with its sequence of chance nodes being evaluated by alternating between levels two and level three. The agent then chooses the branch with the greatest value. Notice that evaluation of the plan tree requires a belief model that is only as deep as the number of steps in the plan. This allows infinite concepts like mutual belief to be represented within a finite model. At each step of evaluation, the planner's belief revision module is invoked on the belief model. For each dialogue act, preconditions and effects are added to every second level of the belief model, according to the precondition and effect rules contained in the beliefs at the level of the evaluating agent. This is a lazy but easy approach, since the revising agent only revises his beliefs about the other agent - he doesn't go so far as to update his private beliefs, and so does not have to resolve conflicts between each agent's beliefs. That is why every second level is updated, rather than every level. Neither does the agent make any further inferences or maintain the consistency of his belief set once revisions are made. This lazy approach has worked well for all of our example problems, but there are dialogues in which it would be useful to make deeper inferences, such as correction of misconception.

3 Negotiation acts

In addition to the domain level acts, which are supplied in the planner's input, the system has a set of built-in negotiation speech acts, which are used to exchange information before the domain level plan is executed. These acts result in a domain level plan that has a higher expected utility, through the agent revising his beliefs. Negotiation planning is a lot like domain-level planning, in that the negotiation acts are part of the game tree and go through the evaluation process in the same way. In fact the domain plan is attached at the leaves of the negotiation plan. However, only some of the negotiation acts can be specified using our plan rules and belief revision mechanism. Others require special more sophisticated belief revision operations to define them. Each is explained in turn.

3.1 "pass"

Pass is intended to allow an agent to pass the turn in a dialogue without saying anything. As such it has no preconditions and no effects. We have given pass a cost of 4, since in a dialogue there is usually some short utterance or a moment of silence before a pass can be inferred. It can be specified using the following plan rule:

```
name:          pass
parameter:     {}
precondition:  {}
effects:       {}
decomposition: {}
```

3.2 "tell"

Tell is intended to allow an agent to inform another agent about a proposition. There is a form of tell for when the proposition is believed, and a form for when its negation is believed. tell-true has a precondition that the agent believes the proposition, and tell-false has a precondition that the agent believes its negation. Using the standard belief revision mechanism, the hearer performs a belief revision step in response to tell, adopting the belief that the precondition held for the speaker, and thus the belief is transferred. Tell has been given a uniform cost of 10.

```
name:           tell-true
parameter:      P
precondition:   bel(P)
effects:        {}
decomposition:  {}

name:           tell-false
parameter:      P
precondition:   bel(not(P))
effects:        {}
decomposition:  {}
```

3.3 "ask"

Ask has two pragmatic forms. ask-forced is representative of requests for information, for which the hearer's response is always to provide it without question. ask-forced has a cost of 10. It is represented by the following plan rules:

```
name:           ask-pair
parameter:      P
precondition:   {}
effects:        {}
decompostion:   { ask-forced; reply(P) }

name:           reply
parameter:      P
precondition:   {}
effects:        {}
decomposition:  { tell-true(P) },
                { tell-false(P) }
```

Ask provides another good illustration of the workings of the planner. The speaker plans an ask-pair, decomposes it, and produces the first act, ask-forced. This forms the single branch of the root node of the game tree. The hearer observes the ask-forced, and realising that the ask-forced node constitutes a full subtree, uses beliefs at level three to infer its parent, ask-pair, which can then be expanded, using level two beliefs, to a reply act. Then, reply is decomposed, giving two alternatives, tell-true and tell-false. Since each of these has a precondition which depends on the belief of the acting agent, a chance node is inserted into the game tree. In the true branch of this chance node, reply can be decomposed only to tell-true, and in the false branch, only to tell false. This ask game tree is shown in the upper branch of the tree in figure 3.

The second pragmatic form of ask is ask-auto, which allows the hearer to reply only if it is rational for him. An English example would be "I would like to know P". The hearer is only required to revise his beliefs to accommodate the fact that the speaker "would like to know", and then decide to answer in the context of this revised model. Unfortunately, there is no simple way to revise the hearer's beliefs, since there could be many different reasons that motivate the speaker. For example, an agent might ask whether there is fruit because he believes that fruit salad is a good candidate plan, or because he prefers to paint a still life. Each explanation requires revision of different beliefs. To cope with this problem, we use a search algorithm, which searches the belief space for a state in which the speaker's asking is rational, but is at the same time as close as possible to the current belief state. The space is treated as a Euclidean space with a dimension for each belief in the belief model. While crude, this mechanism turns out to be effective, in that it produces worthwhile dialogues.

3.4 "propose"

Propose is used by agents to express their preferences over plans, such as in "I would choose P". Like ask-auto, the hearer responds by revising his beliefs, but there is the problem of many explanations that would cause the agent to prefer the plan. Once again, the agent chooses the explanation that has the smallest Euclidean distance from the current belief state. This definition of propose is more useful than that given by Gmytrasiewicz and Durfee [2001]. In their model, the agent merely prunes the game tree. In our model, beliefs are updated instead, with the side effect that the desired branch in the game tree is selected. Propose has a cost of 10.

3.5 Selection of the repertoire

We cannot claim that the set of acts produces the most efficient dialogues, but it seems unlikely that there would be any other acts as simple as these. The repertoire covers the obvious basic units of the agent's mental state, namely his beliefs and preferences, and covers all of the simple pragmatic definitions that are possible within the bounds of the belief revision mechanism. They also correspond well with acts seen in human collaborative planning, such as in the TRAINS [Allen, 1995] corpus, and those that appear in speech act theory and work on communication languages in artificial multi-agent systems. While perhaps not efficient, it is clear that they can eventually produce the most efficient domain plan, since using tell alone on every belief will lead each agent to a perfect model of the other.

One act that we haven't included is request, which is to propose as ask-forced is to ask-auto, in that it obligates the hearer to act. However, propose is already quite powerful in that the second agent searches the belief space from level three upwards to accommodate the proposed plan. For request to dominate propose, request must demand revisions at level two as well. However, this violates our assumption of lazy belief revision in which the second agent never revises his private beliefs. We suspect though that request would be useful with a less lazy belief revision mechanism.

4 Experiments

We have exercised the planner on a series of dialogue problems intended for use with a kitchen-assistant-robot, who

must coordinate his plan with the user so that they harmoniously prepare a meal together. These experiments are intended to show primarily that each of the negotiation acts in the agent's repertoire is a necessary member, in that it dominates all of the others in some examples. They also characterise some common decision problems seen in negotiation dialogues, and show how these decisions are subtly dependent on probabilistic values in the belief model, rather than on logical values.

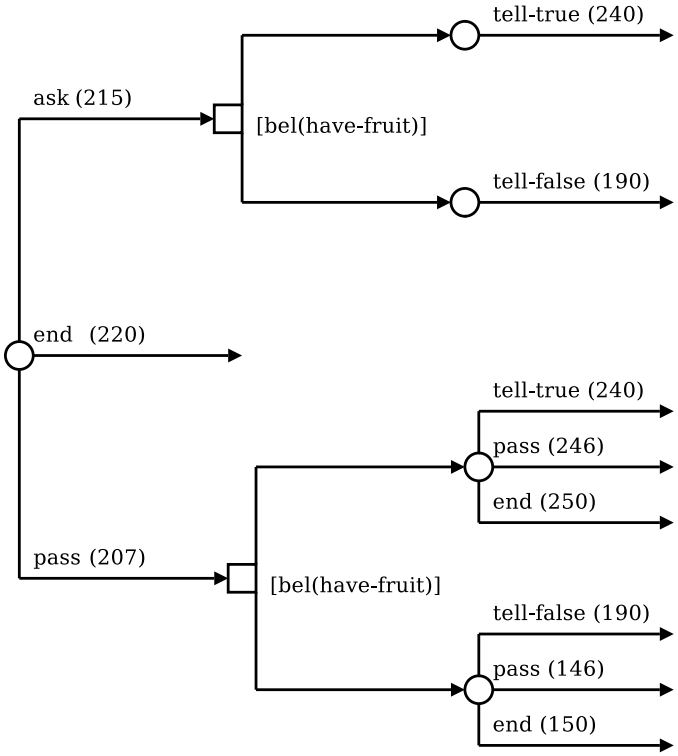

Figure 3: Game Tree in Experiment One

4.1 Experiment One

Experiment 1 demonstrates the competition of initiative between asking a question and waiting to be told, in the context of the have-fruit problem introduced earlier. Asking is more expensive than waiting since it involves an ask act and a reply act, both of which cost 10 units. On the other hand, the agent can pass at a cost of 4 and risk that the second agent will decide to tell him the answer without being asked, at a cost of 10. At a total of 14 units this costs less, but it is risky. The planner generates the game tree in figure 3 for the dialogue, which for illustration is evaluated at the point [0.7,0.7] at levels [2,4] in a belief space which represents the belief have-fruit at levels 2, 4, 6, and so on.

The overall efficiency of each strategy is plotted in figure 4, with the level two belief along the left axis, and the level four belief along the right axis. Notice that the belief needs to be evaluated at both levels two and four to make the decision, and that all three strategies of asking, waiting and proceeding

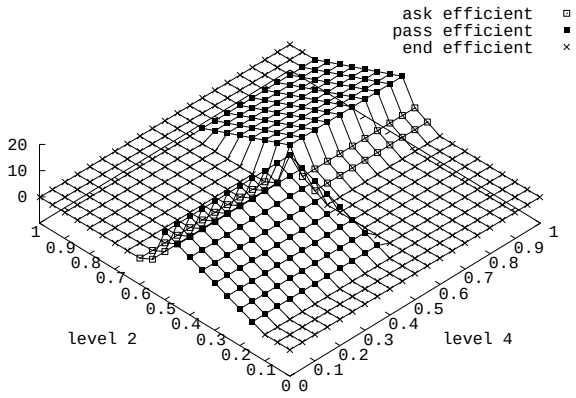

Figure 4: Utility of Strategies in Experiment One

directly to the domain-level plan are dominant in different regions of the belief space. It is clear from examining each strategy's curve that a probabilistic approach must be taken to modelling the agent's beliefs. There are sharp decision surfaces across which utility changes dramatically, and so an inaccurate belief model can make a significant difference to the performance of the system. Notice as well that there appears to be no simple rule, short of evaluating the game tree using both levels two and four, that would be satisfactory to solve this problem.

4.2 Experiment Two

In experiment two we introduced a second variable which modifies the efficiency of waiting to be told. A precondition "have-eggs" is used with the make-omelette strategy, giving the game tree in figure 5 The result is that if the second agent believes there are no eggs, he will not bother to tell the first agent that he has fruit, even though the first agent privately believes that he has eggs. This demonstrates that the relative efficiency of asking and waiting to be told can vary. In fact, below a certain threshold, telling becomes always inefficient for the second agent. This is just a small example of the general phenomenon of competition for negotiation between different plans in a larger game tree. If a chance node appears near the root, the value of each of its branches with respect to each agent makes all the difference in deciding whether a plan attached to those branches is worth negotiating. As a result, agents try to either hold or decline the floor with respect to negotiation topics. In this instance, the first agent sees that his plan is more important than it is perceived, and aggressively takes the floor by asking.

4.3 Experiment Three

Experiment three investigates the propose act, showing how it can be more efficient than an equivalent set of tell acts, or any other combination of acts. Propose is most useful when it is communicating a choice that is unexpected by the other agent, since by doing so, many beliefs can be significantly revised using just one act.

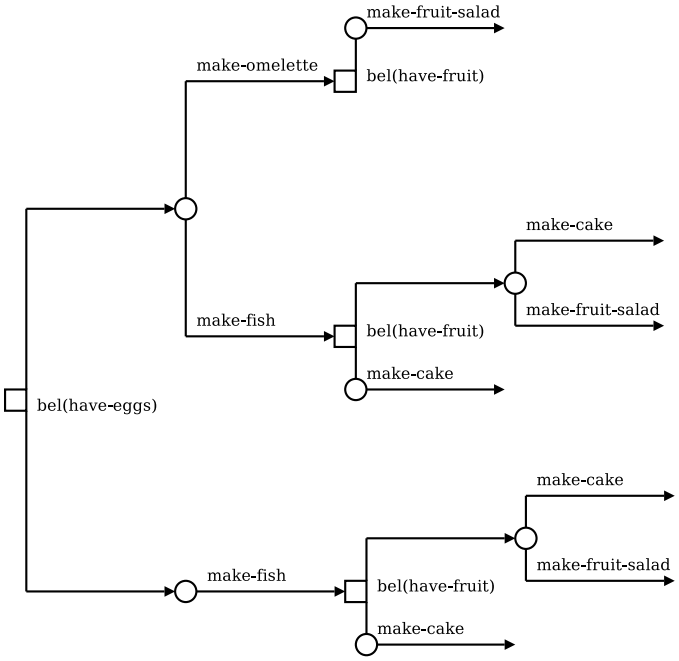

Figure 5: Game Tree for Experiment Two

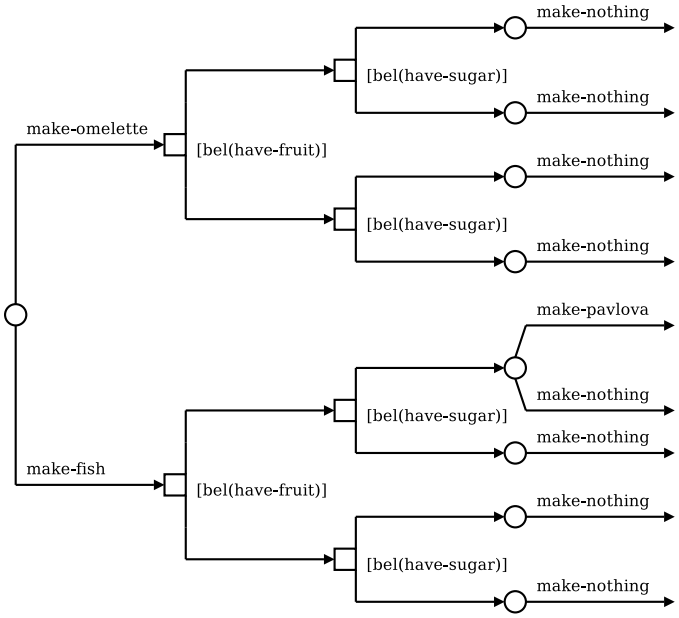

Figure 6: Game Tree for Experiment Three

We constructed a problem in which the second agent would prefer to make a pavlova (figure 6). Unfortunately, he will have to tell the first agent both that he has sugar, and that he has fruit if the first agent is to be convinced to choose make-fish, and leave the eggs for the pavlova. This is relatively expensive. The same effect can be achieved by simply proposing to make a pavlova. Then, the belief revision mechanism revises the hearer beliefs to a state in which both of the beliefs are high enough that the proposal is accommodated.

Negotiation acts were added to the agent's repertoire one by one to demonstrate the utility gain offered by each. To start, there were no negotiation acts, and so the game tree just consisted of the domain-level tree, with a value of 100 for make-omelette. Next, the pass and tell acts were added. This produced the negotiation game tree in the upper part of figure 8. This tree shows the best strategy only for the agent, so that each choice node is pruned down to only one alternative. Notice that in response to a pass, the second agent uses a pair of informs in the **true,true** branch of the game tree. This subdialogue is efficient, and since it happens in one quarter of instances, the value for the tree is 102.5, which is a marginal gain over the 100 obtained from the plain domain-level plan. Next, the ask acts were added, but these were dominated by the pass and inform combination, and so the same result of 102.5 was obtained. Next, propose was added. This produced the tree in the bottom part of figure 8, with propose dominating instead of tell. Since the negotiation ends with the proposal, there is a smaller cost than in the upper game tree in figure 8. The overall cost of the dialogue turns out to be 106, compared with 102.5 obtained without using propose.

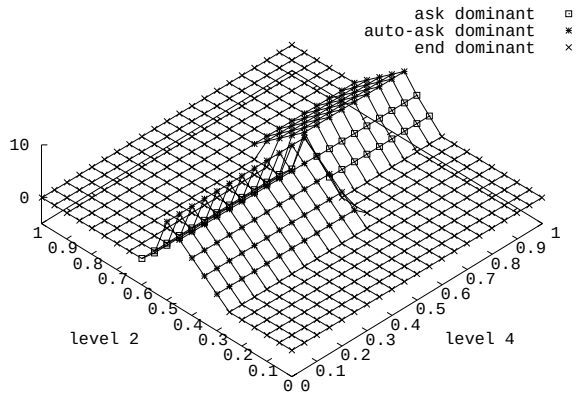

Figure 7: Dominance of ask-auto over ask-forced

4.4 Experiment Four

In experiment four, we compared the two pragmatic senses of ask, to demonstrate that both are required for efficient dialogue. We took the problem that was used in experiment two, where it happened that waiting to be told was never efficient, but asking was. Ask-auto causes the second agent to revise his beliefs about the eggs in the search for a belief state in which asking would be efficient for the first agent. After that, the agent uses a tell, which is now efficient because of the eggs revision. ask-auto therefore has a similar character to waiting for a tell, but is a little less efficient due to the cost of having to ask. The reason that ask-auto has an advantage over ask-forced is that ask-auto uses the level 4 belief in making the tell decision, whereas ask-forced only uses the level 2 belief. Figure 7 shows that both ask-forced and ask-auto are dominant at different times, whereas waiting for a tell is

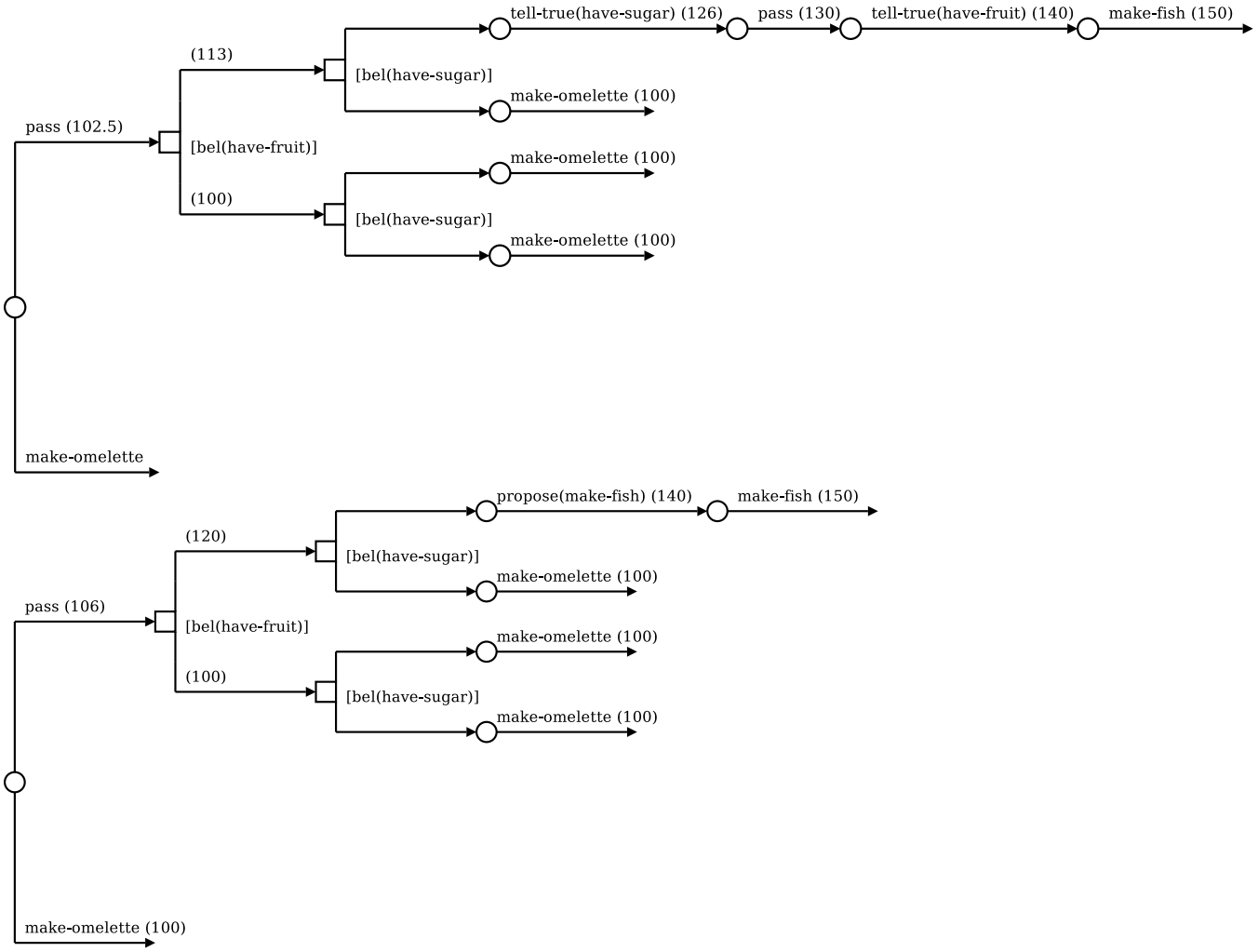

Figure 8: Negotiation Game Trees for Experiment 3

never dominant. Therefore each sense of asking is included in the repertoire.

5 Conclusion

We have described a planner that uses game trees and a probabilistic user model to produce adaptive dialogue strategies. We described a repertoire of negotiation acts - pass, ask, tell, and propose, each of which complements the others. The use of a probabilistic model was shown to be important to achieving efficient dialogue, since the utility of the strategies varies across a probabilistic belief space.

References

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

[Allen, 1995] James F. Allen. The TRAINS project: A case study in building a conversational planning agent. *Journal of Experimental and Theoretical AI (JETAI)*, 7:7–48, 1995.

[Carberry, 1990] Sandra Carberry. *Plan Recognition in Natural Language Dialogue*. MIT Press, 1990.

[Carletta, 1992] Jean Carletta. Planning to fail, not failing to plan: risk-taking and recovery in task-oriented dialogue. In *Proceedings of the 14th conference on Computational linguistics*, pages 896–900. Association for Computational Linguistics, 1992.

[Gmytrasiewicz and Durfee, 2001] Piotr J. Gmytrasiewicz and Edmund H. Durfee. Rational communication in multi-agent environments. *Autonomous Agents and Multi-Agent Systems*, 4(3):233–272, 2001.

[Grosz and Sidner, 1986] Barbara J. Grosz and Candace L. Sidner. Attention, intentions, and the structure of discourse. *Comput. Linguist.*, 12(3):175–204, 1986.

[Heeman and Hirst, 1995] Peter A. Heeman and Graeme Hirst. Collaborating on referring expressions. *Comput. Linguist.*, 21(3):351–382, 1995.

[McCoy, 1989] K. F. McCoy. Highlighting a user model to respond to misconceptions. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, pages 233–254. Springer, Berlin, Heidelberg, 1989.

[McEleney and O'Hare, 2005] Bryan McEleney and Gregory O'Hare. Decision theoretic planning for initiative problems. In *Proceedings of the 10th International Conference on User Modelling*, Edinburgh, Scotland, September 2005. Springer Verlag.

[Pollack, 1986] Martha E. Pollack. A model of plan inference that distinguishes between the beliefs of actors and observers. In *Proceedings of the 24th conference on Association for Computational Linguistics*, pages 207–214. Association for Computational Linguistics, 1986.

[Sacerdoti, 1974] E.D. Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, 5:115–135, 1974.