An Approach to Automated Agent Negotiation using Belief Revision

Pablo Pilotti[1], Ana Casali[1,2] and Carlos Chesñevar[3]

[1] Centro Internacional Franco Argentino de
Ciencias de la Información y de Sistemas (CIFASIS) Rosario,
Av. 27 de febrero 210 bis - S2000EZP, Rosario, Argentina
Email: pilotti@cifasis-conicet.gov.ar

[2] Facultad de Cs. Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario (UNR)
Av. Pellegrini 250 - S2000BTP, Rosario, Argentina
Email: acasali@fceia.unr.edu.ar

[3] Depto. de Cs. e Ingeniería de la Computación
Universidad Nacional del Sur (UNS) - CONICET
Av. Alem 1253 - B8000CPB Bahía Blanca, Argentina
Email: cic@cs.uns.edu.ar

Abstract. A typical scenario for negotiation involves agents which cannot reach their goals by themselves because they do not have some resources or they do not know how to use them to reach their goals. Also, they may have incomplete or wrong information about the other agent's goals and resources. In this paper we present an approach to automated negotiation based on belief revision operators, where agents offer resources and plans for exchanging in order to achieve their goals. This approach is presented through a high-level algorithm, formalized in COQ and implemented in logic programming. As a case study, we show how the well-known hammer-nail-mirror problem can be solved in the context of our proposal.

1 Introduction

Negotiation is a form of interaction in which two or more agents with different goals find some acceptable agreement. A typical scenario for negotiation involves two agents who have the need to collaborate for mutual benefit. Automated negotiation research can be considered to deal with three broad topics [5]: *a) Negotiation Protocols*, the set of rules that govern the interaction; *b) Negotiation Objects*, the range of issues over which agreement must be reached and *c) Agents' Decision Making Models*, the decision making apparatus the participants employ to act in line with the negotiation protocol in order to achieve their objetives.

In this paper, we focus on a particular case of negotiation between two agents (Ag_i and Ag_j) that can not reach their goals alone and ask for help to the other.

They also may have incomplete or wrong information about the other agent's goals and resources. As motivational example we work on a slightly modified version of the HNM example [6]:
The Agent Ag_1 has as goal hanging a picture, and it has a screw, a screwdriver, a hammer. Also, he knows how a hammer and a nail can be used to hang a picture and how a screw and a screwdriver can be used to hang mirrors. On the other hand, Agent Ag_2 has as goal to hang a mirror, and it has a nail and the knowledge of how to hang a mirror using a hammer and a nail. Neither Ag_1 nor Ag_2 can reach their goals on the basis of their knowledge and resources; they need to perform some exchanges in order to do so.

We present an approach which is based on belief revision where agents' beliefs are updated as a negotiation proceeds, resulting in different moves captured in a dialogue. The algorithm proposed for the agents decision making apparatus generates automatically those proposals corresponding to suitable exchanges of resources and plans, which can lead both agents to achieve their goals. The proposals the agents make as the dialogue takes place are the ones they believe closer to a final agreement and are based on the beliefs they have about each other. The presented approach was formalized in COQ[4] and the high-level algorithm proposed for negotiation has been implemented in logic programming. As a case study, we show how the HNM problem can be solved in the context of our proposal.

The remainder of this paper is structured as follows: in Section 2 we define the agent architecture and the negotiation protocol, formalizing the notions of proposal, dialogue and deal. In Section 3, we propose a high-level algorithm for solving negotiation problems between two agents, based on belief revision operators. Next, in Section 4 we show how the HNM problem can be solved in the context of our computable model. Finally, we discuss conclusions and related work in Section 5.

2 Agent Architecture and Negotiation Protocol

Let $\mathcal{L}$ denote a propositional language, in which the following subsets are distinguished:

- $R_{\mathcal{L}}$: a set of atoms representing objects which stand for resources for an agent (e.g. $nail$, $hammer$).
- $G_{\mathcal{L}}$: a set of atoms representing goals (ex. $hangMirror$, $hangPicture$).
- $P_{\mathcal{L}}$: a set of propositional formulae encoding plans, which may involve objects for achieving a goal. (e.g. $nail \wedge hammer \rightarrow hangPicture$).

Definition 1. *The mental state of an Agent Ag_i, is a tuple $MS_i = <R_i, P_i, G_i, BR_i, BP_i, BG_i>$, where: $R_i, BR_i \subset R_{\mathcal{L}}$; $P_i, BP_i \subset P_{\mathcal{L}}$; and $G_i, BG_i \subset G_{\mathcal{L}}$.*

Thus, each agent's mental state has a set of available resources (R_i), plans (P_i), and a set of goals to achieve (G_i), as well as belief sets about which resources

[4] http://coq.inria.fr/

and plans are available for agent Ag_j (BR_i and BP_i), and a set BG_i of beliefs about which goals are those of agent Ag_j.

Based on their mental states, the agents will generate proposals towards reaching their goals. In our work, a proposal is a statement that includes what the agent want to receive, together with an explanation justifying why an agent needs what he is asking for, and what the agent would give in return. Thus, proposals will have the following intended meaning:

`I propose that you provide me` R_{get}`, because if I use` R_{own}`, then I can achieve` G `in exchange for` R_{give}`.`

where R_{get}, R_{own}, and R_{give} stand for resources and plans, and G is a set of goals.

Definition 2. *Let R_{get}, R_{own}, and R_{give} subsets of $R_{\mathcal{L}} \cup P_{\mathcal{L}}$, and G subset of $G_{\mathcal{L}}$. A* proposal *is a 4-tuple $(R_{get}, R_{own}, G, R_{give})$ such that:*

$$R_{get} \cup R_{own} \vdash G \tag{1}$$

$$R_{own} \nvdash G \tag{2}$$

$$R_{give} \cap (R_{get} \cup R_{own}) = \emptyset \tag{3}$$

Notice that (1) states that both sets of resources and plans R_{own} and R_{get} are needed for the agent to reach the goal G; (2) means that without the required set R_{get} of resources and plans the agent can not reach the goal G and (3) states that the set of resources and plans R_{give} is not needed by the agent to reach G.[5].

A dialogue between two agents consists of a finite sequence of proposals (performed alteratively by each of the agents involved in the dialogue), ending with *accept* (there is a deal) or *withdraw* (no deal is possible).[6]

Definition 3. *A* dialogue *between agents Ag_i and Ag_j is a finite sequence of utterances $[u_1, ..., u_{n-1}, u_n]$ where for $r < n$, u_r is a proposal and $u_n \in \{accept, withdraw\}$, such that: (1) there are no repeated utterances, i.e. $u_s \neq u_t$, with $t, s < n$; (2) utterance u_k with $k > 1$ is performed by Agent Ag_i only if utterance u_{k-1} is performed by Agent Ag_j (i.e. agents alternate moves). A dialogue will be* initiated *by Ag_i iff u_1 is performed by Ag_i.*

Note that dialogues can be warranted to be finite, as there is a finite set of possible combinations of proposals and utterance repetition is not allowed.

Definition 4. *The decision making apparatus of an Agent Ag_i is a tuple $D_i =< History_i, Init_i, Answer_i >$, where*

$History_i$ *is the negotiation dialogue.*
$Init_i : MS_i \times History \rightarrow Proposal \times History$
$Answer_i : MS_i \times History \times Proposal \rightarrow$
$\quad MS_i \times History \times Proposal \cup \{accept, withdraw\}$

[5] We write $X \vdash G$ whenever $G \subseteq Cn(X)$, where Cn is a logical consequence operator.
[6] The formal definition of deal is given in Section 2.1.

Definition 5. *An agent Ag_i is a tuple $< MS_i, D_i >$, where MS_i is its mental state and D_i its decision making apparatus.*

A dialogue between agents Ag_i and Ag_j will be started by one of the agents with a proposal computed by $Init$, followed by a counter-proposal by the other agent computed by $Answer$, a counter-counter-proposal by the first agent, and so on. Without loss of generality we assume the agent Ag_i starts the negotiation. Figure 1 represents the negotiation dialogue flow initiated by Ag_i as a finite-state machine.

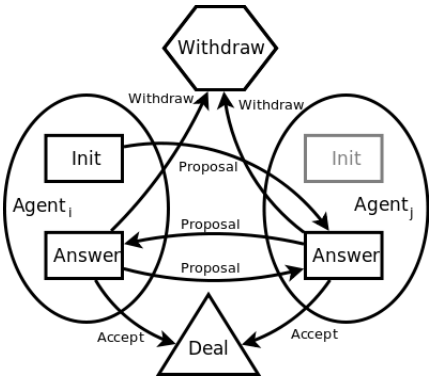

Fig. 1. Dialogue flow initiated by Ag_i

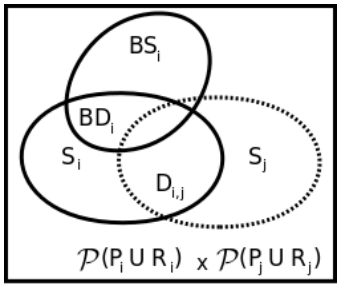

Fig. 2. Solutions' space from Ag_i viewpoint

2.1 Agent deals

We assume agents Ag_i and Ag_j cannot reach their objectives by their own, (i.e. $\forall k \in \{i, j\}, R_k \cup P_k \nvdash G_k$), and therefore the problem each agent faces is to find a suitable exchange of plans and resources in the space of possible ones ($\mathcal{P}(P_i \cup R_i) \times \mathcal{P}(P_j \cup R_j)$) in order to reach his own goal. Clearly, each agent is aware of his own resources, plans and goals, and he may also have beliefs of the other agent's mental state (resources, plans and goals). Thus, from his viewpoint he can determine which are the exchanges corresponding to solutions.

Definition 6. *Let Ag_i be an agent involved in a negotation. A* solution for Ag_i *(noted by $\mathcal{S}_i$) is any pair (X, Y), $X, Y \subseteq R_{\mathcal{L}} \cup P_{\mathcal{L}}$ such that: 1) $X \subseteq (R_i \cup P_i)$ and 2) $((R_i \cup P_i) - X) \cup Y \vdash G_i$*

In a similar way $\mathcal{S}_j$ is defined. Note that X stands for those resources and plans that Ag_i is willing to give to Ag_j, whereas Y is the set of resources and plans that are given to Ag_i to achieve his goal. A *deal* for Ag_i and Ag_j will be a solution which is applicable for both of them. Formally:

Definition 7. *We will say that* (X, Y) *where* $X, Y \subseteq R_{\mathcal{L}} \cup P_{\mathcal{L}}$*, is a* deal *for* Ag_i *and* Ag_j *iff* $(X, Y) \in \mathcal{S}_i \cap \mathcal{S}_j$*. We will denote with* $\mathcal{D}_{i,j}$ *the set of all deals between* Ag_i *and* Ag_j*.*

According to Def. 2, a proposal for an agent Ag_i is a tuple (R_{get}, R_{own}, G, R_{give}). Clearly, the pair of resources (R_{give}, R_{get}) provides a solution to reach Ag_i's goal, i.e. $(R_{give}, R_{get}) \in \mathcal{S}_i$. We define the function $\odot$ that assigns to each proposal (R_{get}, R_{own}, G, R_{give}) its *associated solution*. The beliefs a particular agent has about the other agent's goals, resources and plans in a negotiation dialogue are significant as they can help reaching a deal. From this information, an agent can infer which proposals he believes that are more suitable for the other, and consequently more likely to be accepted. To formalize this notion we define the following concepts.

Definition 8. *Let* Ag_i *and* Ag_j *be two agents, we will say that* Ag_i believes (X, Y) *is a* solution *for* Ag_j *whenever: 1)*$Y \subseteq (BR_i \cup BP_i)$ *and 2)* $(BR_i \cup BP_i - Y) \cup X \vdash BG_i$*. We will define* $\mathcal{BS}_i = \{(X, Y) \mid Ag_i$ believes (X, Y) *is a* solution *for* Ag_j $\}$*.*

Definition 9. *Let* Ag_i *and* Ag_j *be two agents, we will say that* Ag_i believes (X, Y) *is a* deal *for* Ag_j *iff: 1)* $X \subseteq (R_i \cup P_i)$*, 2)* $Y \subseteq (BR_i \cup BP_i)$*, 3)* $(R_i \cup P_i - X) \cup Y \vdash G_i$ *and 4)* $(BR_i \cup BP_i - Y) \cup X \vdash BG_i$*. We will define* $\mathcal{BD}_i = \{(X, Y) \mid Ag_i$ believes (X, Y) *is a* deal *for* Ag_j $\}$*.*

From definitions 8 and 9 the following propositions hold:[7]

Proposition 1. $(X, Y) \in \mathcal{S}_i$ *and* $(X, Y) \in \mathcal{BS}_i \Leftrightarrow (X, Y) \in \mathcal{BD}_i$*.*

Proposition 2. $(X, Y) \in \mathcal{BD}_i$ *and* $(X, Y) \in \mathcal{S}_j \Rightarrow (X, Y) \in \mathcal{D}_{i,j}$*.*

Proposition 3. $(X, Y) \in \mathcal{BD}_i$ *and* $(X, Y) \in \mathcal{BD}_j \Rightarrow (X, Y) \in \mathcal{D}_{i,j}$*.*

Proposition 1 states that if a pair (X, Y) is solution for Ag_i and he believes that it is also a solution for Ag_j, then Ag_i believes that (X, Y) is a deal. Similarly, Proposition 2 asserts that if the agent Ag_i believes that (X, Y) is a deal and (X, Y) is also a solution for Ag_j, then (X, Y) is a deal. Finally, Proposition 3 states that if both agents believe that (X, Y) is a deal, then it holds that (X, Y) is a deal.

Figure 2 shows the set of solutions from the viewpoint of Ag_i. The dotted line represents that the agent does not know $\mathcal{S}_j$ with certainty. Because of this, he can not be sure of making a *proposal* such that $\odot proposal \in \mathcal{D}_{i,j}$. Our approach is based on the following intuition: in order for agent Ag_i to reach a deal in the negotiation process, he will revise his believes in order to have his potential solutions $\mathcal{BS}_i$ as close to $\mathcal{S}_j$ as possible, and consequently the resulting set of possible deals $\mathcal{BD}_i$ will be closer to $\mathcal{D}_{i,j}$ as well. In this scenario we assume that since agents need to be colaborative, they are"truthful". Therefore, if Ag_i revises his beliefs from the dialogue information, it would be a good strategy for him to prioritize the pairs $\mathcal{BD}_i$ at the time of generating proposals.

[7] All the propositions presented and their proofs were developed in COQ and can be seen in `http://web.cifasis-conicet.gov.ar/~pilotti/Automated_Agent_Negotiation.v`

3 Integration Belief revision and Negotiation

Classic belief change operations introduced in the AGM model [1] are known as *expansions*, *contractions* and *revisions*. An expansion incorporates a new belief without warranting the consistency of the resulting epistemic state. A contraction eliminates a belief α from the epistemic state as well as all those beliefs that make the inference of α possible. Finally, a revision incorporates a new belief α to the epistemic state warranting a consistent result, assuming that α itself is consistent.

As discussed before, in our setting we assume that the agents have their own beliefs about the other agent's resources, plans and goals. It must be noted that the sets of resources, plans and objectives *do not* change during the negotiation; only if a deal succeeds at the end of the negotiation process, the actual exchange of resources and plans will take place (and consequently the sets R_i, P_i, R_j and P_j will be changed). In order to model such a negotiation process in terms of belief revision we will use the notion of *Choice kernel Set* and *Multiple Choice contraction* proposed by Hansson [4] and followed by Fermé et al [3]. These notions will be useful for providing a practical approach to belief revision in our context. In order to make this paper self-contained, we provide below the formal definitions involved.

Definition 10 ([3]). *Let $\mathcal{L}$ be a logical language, Cn a consecuence operator, $K \subseteq \mathcal{L}$ and $A \in \mathcal{L}$. Then $K \perp\!\!\!\perp A$ is the set of all X such that a) $X \subseteq K$; b) $A \subseteq Cn(X)$, and c) if $Y \subset X$ then $A \nsubseteq Cn(Y)$. The set $K \perp\!\!\!\perp A$ is called* Choice kernel Set*, and its elements are called* A-kernels of K.

Informally, a Choice kernel Set is a minimal belief subset of the epistemic state from which A can be deduced. *Incision functions* cut into each A-kernel, removing at least one sentence from them. Since all A-kernels are minimal subsets implying α, from the resulting sets it is no longer possible to derive α.

Definition 11 ([3]). *A function σ is a incision function σ for K, iff satisfies for all A: a) $\sigma(K \perp\!\!\!\perp A) \subseteq \bigcup(K \perp\!\!\!\perp A)$ and b)If $\emptyset \neq X \in K \perp\!\!\!\perp A$, then $X \cap \sigma(K \perp\!\!\!\perp A) \neq \emptyset$*

Definition 12 ([3]). *Let σ be an incision function for K and $A \in \mathcal{L}$. The* multiple choice contraction $\approx$ *for K is defined as follows: $K \approx A = K - \sigma(K \perp\!\!\!\perp A)$*

Definition 13 ([4]). *Let $\approx$ be a global kernel contraction. Given a set of sentences K, we define for any set A the revision operator $*$: $K * A = (K \approx \neg A) \cup \{A\}$*

Suppose that Ag_i makes the proposal $(R_{get}, R_{own}, G, R_{give})$ to Ag_j. As in our approach the agents are truthful, the agent Ag_j can infer from the received proposal the following information:

1. If Ag_i asks for R_{get} then $(R_{get} \cap R_{\mathcal{L}}) \nsubseteq R_i$ and $(R_{get} \cap P_{\mathcal{L}}) \nsubseteq P_i$.

2. If Ag_i uses R_{own} then $(R_{own} \cap R_{\mathcal{L}}) \subseteq R_i$ and $(R_{own} \cap P_{\mathcal{L}}) \subseteq P_i$.
3. If Ag_i wants to reach G then $(G \cap G_{\mathcal{L}}) \subseteq G_i$.
4. If Ag_i offers R_{give} then $(R_{give} \cap R_{\mathcal{L}}) \subseteq R_i$ and $(R_{give} \cap P_{\mathcal{L}}) \subseteq P_i$.

Then, from the infered information Ag_j can update his beliefs through the following steps which can be seen as variable assignments:

1. $BR_j \leftarrow BR_j \approx R_{get} \cap R_{\mathcal{L}}$
2. $BR_j \leftarrow BR_j * R_{own} \cap R_{\mathcal{L}}$
3. $BR_j \leftarrow BR_j * R_{give} \cap R_{\mathcal{L}}$
4. $BP_j \leftarrow BP_j \approx R_{get} \cap P_{\mathcal{L}}$
5. $BP_j \leftarrow BP_j * R_{own} \cap P_{\mathcal{L}}$
6. $BP_j \leftarrow BP_j * R_{give} \cap P_{\mathcal{L}}$
7. $BG_j \leftarrow BG_j * G \cap G_{\mathcal{L}}$

3.1 The Agent Decision Model: High-level algorithms

As stated earlier, each agent's decision model has been implemented by using two algorithms $Init$ and $Answer$. The algorithm $Init$ is in charge of starting the negotiation. In a first place, it selects a proposal that the agent Ag_i believes is a deal ($\mathcal{BD}_i$) that has not been proposed before. If such proposal does not exist, it tries to send a proposal associated with his own solutions ($\mathcal{S}_i$). If it fails, the agent sends a withdraw message. On its turn, $Answer$ receives the proposal generated from $Init$ and checks if it is an associated solution to the agents problem, and in that the proposal is accepted. If that is not the case, the agent's beliefs are revised and $Init$ is called to generate a new proposal. High-level algorithms for $Init_i$ and $Answer_i$ are given next.

Algorithm 1 : In line 1, the function Gen_i is used (see Def. 14) to compute the set of proposals $prop\mathcal{S}_i$ such that their associated solutions belong to $\mathcal{S}_i$. Similary, in line 2, Gen_i is used to compute the set of proposals $prop\mathcal{BS}_i$ that the agent believes their associated solutions belong to $\mathcal{BS}_i$. In line 3, the set $prop\mathcal{BD}_i$ is computed as those proposals in $prop\mathcal{S}_i$ such that their associated solutions are potencial deals (see Prop 1). In line 4, those proposals that have been offered before are discarded. The *select* function chooses one proposal out of the set *propSet* of possible candidate proposals. [8] Finally, the selected *prop* is added to the $History$.

Algorithm 2 : In lines 1-2, the $History$ is updated, and the set $prop\mathcal{S}_i$ is computed. In line 3, we check if the solution associated with the received proposal is a solution for Ag_i. For this purpose, we use $\odot$ to denote the associated solution of given proposal and $\bigodot$ to denote the set of associated solutions of a set of proposals. Then, in lines 6 to 13, the agent updates his mental state following the steps presented in Section 3 by means of the functions $RES()$ and $PLA()$ (which return resources and plans in a given set, resp.).

[8] We abstract away this selection function, which could be defined according to some valuation criterion (e.g. cost).

Algorithm 1: $Init_i$
Input: $MS_i, History$
Output: $Proposal, History$

1: $propS_i \leftarrow \mathsf{Gen}_i(R_i \cup P_i, BR_i \cup BP_i, G_i)$
2: $propBS_i \leftarrow \mathsf{Gen}_i(BR_i \cup BP_i, R_i \cup P_i, BG_i)$
3: $propBD_i \leftarrow propS_i \ominus propBS_i$
4: $propSet \leftarrow propBD_i - sent_i(History)$
5: **if** $propSet \neq \emptyset$ **then**
6: $prop \leftarrow select(propSet)$
7: $add(History, prop)$
8: **return** $prop$
9: **else**
10: $propSet \leftarrow propS_i - sent_i(History)$
11: **if** $propSet \neq \emptyset$ **then**
12: $prop \leftarrow select(propSet)$
13: $add(History, prop)$
14: **return** $prop$
15: **else**
16: **return** $withdraw$
17: **end if**
18: **end if**

Algorithm 2: $Answer_i$
Input: $MS_i, History, Proposal$
Output: $MS_i, History, Proposal$

1: $add(History, prop)$
2: $propS_i \leftarrow \mathsf{Gen}_i(R_i \cup P_i, BR_i \cup BP_i, G_i)$
3: **if** $\odot(prop) \in \bigodot(propS_i)$ **then**
4: **return** $accept$
5: **else**
6: $BR_i \leftarrow contract(BR_i, RES(R_{get}))$
7: $BR_i \leftarrow revise(BR_i, RES(R_{own}))$
8: $BR_i \leftarrow revise(BR_i, RES(R_{give}))$
9: $BP_i \leftarrow contract(BP_i, PLA(R_{get}))$
10: $BP_i \leftarrow revise(BP_i, PLA(R_{own}))$
11: $BP_i \leftarrow revise(BP_i, PLA(R_{give}))$
12: $BG_i \leftarrow revise(BG_i, G)$
13: $prop \leftarrow Init_i(MS_i, History)$
14: **return** $prop$
15: **end if**

Note that the function Gen is used to compute the proposals that are solution to the Ag_i problem (i.e. $\odot prop \in \mathcal{S}_i$), and using diferent arguments, it is also used to compute proposals that are potencial solutions (i.e. $\odot prop \in \mathcal{BS}_i$). Below we specify the Gen function and some properties that follow from its specification are given.

Definition 14. *Let $L_{PR}, L_{BPR} \subset P_{\mathcal{L}} \cup R_{\mathcal{L}}$ and $L_G \subset G_{\mathcal{L}}$, we define the function* Gen *as follows:*

$$\mathsf{Gen}(L_{PR}, L_{BPR}, L_G) = \{\ (R_{get}, R_{own}, G, R_{give}) : (R_{own} \cup R_{get}) \in (L_{PR} \cup L_{BPR} \cup R_{get}) \perp\!\!\!\perp L_G, \ R_{get} \cap L_{PR} = \emptyset, \ \ R_{own} \subseteq L_{PR}, \ R_{give} \subseteq L_{PR} - R_{own},\ G = L_G\ \}$$

Proposition 4. *Given an agent Ag_i, where his mental state is $MS_i = <R_i, P_i, G_i, BR_i, BP_i, BG_i>$, then, the following holds:*

1. *If $(R_{get}, R_{own}, G, R_{give}) \in \mathsf{Gen}(R_i \cup P_i, BR_i \cup BP_i, G_i)$*
   (a) *then $(R_{get}, R_{own}, G, R_{give}) \in Proposal$*
   (b) *then $(R_{give}, R_{get}) \in \mathcal{S}_i$ i.e. $\odot(R_{get}, R_{own}, G, R_{give}) \in \mathcal{S}_i$*
2. *If $(R_{get}, R_{own}, G, R_{give}) \in \mathsf{Gen}(BR_i \cup BP_i, R_i \cup P_i, BG_i)$ then $(R_{get}, R_{give}) \in \mathcal{BS}_i$*

4 The Home Improvement Agents Problem revisited

As a case study we use a slightly modified version of the hammer-nail-mirror example [6]. Additionally to the agents' beliefs in the original example we consider the following beliefs: Ag_1 believes that Ag_2 has a *nail* and that his goal is to have an *screw* and Ag_2 believes that Ag_1 has a *nail*. Therefore, Ag_1 and Ag_2 have the following initial states:

$$\begin{aligned} History_1 &= [] \\ R_1 &= \{screw, screwdriver, hammer\} \\ P_1 &= \{hammer \wedge nail \rightarrow hangPicture, \\ &\quad screw \wedge screwdriver \rightarrow hangMirror\} \\ G_1 &= \{hangPicture\} \\ BR_1 &= \{nail\} \\ BP_1 &= \{\} \\ BG_1 &= \{screw\} \end{aligned} \qquad \begin{aligned} History_2 &= [] \\ R_2 &= \{nail\} \\ P_2 &= \{hammer \wedge nail \rightarrow \\ &\quad hangMirror\} \\ G_2 &= \{hangMirror\} \\ BR_2 &= \{nail\} \\ BP_2 &= \{\} \\ BG_2 &= \{\} \end{aligned}$$

Suppose that Ag_1 is the agent that starts the negotiation. Next we summarize the main steps in the first two moves in the negotiation process:

1. Ag_1 uses the algorithm $Init_1$ to compute the first proposal. The functions $\mathsf{Gen}_1(R_1 \cup P_1, \{nail\}, \{hangPicture\})$ and $\mathsf{Gen}_1(\{nail\}, R_1 \cup P_1, \{screw\})$ are computed, obtaining as a result:

$$\begin{aligned} prop\mathcal{S}_1 = \{ &(\{hangPicture\}, \{\}, \{hangPicture\}, R_1), \\ &(\{nail\}, \{hammer, nail \wedge hammer \rightarrow hangPicture\}, \{hangPicture\}, \\ &\{screw, screwdriver, screw \wedge screwdriver \rightarrow hangMirror\}), ...\} \\ prop\mathcal{BS}_1 = \{ &(\{screw\}, \emptyset, \{screw\}, \{nail\}),\ (\{screw\}, \emptyset, \{screw\}, \emptyset)\} \end{aligned}$$

   Now Ag_1 can compute the potential deals from the set of his proposals (i.e. $prop \in prop\mathcal{S}_1$) considering those he believes are solutions for Ag_2 (i.e. $\odot prop \in \bigodot_2 prop\mathcal{BS}_1$):

$$\begin{aligned} prop\mathcal{BD}_1 = &\{(\{nail\}, \{hammer, nail \wedge hammer \rightarrow hangPicture\}, \{hangPicture\}, \\ &\{screw\})\} \end{aligned}$$

   Since this is the first move, $History$ is empty and thus $propSet = prop\mathcal{BD}_i$ is a singleton. Then the *select* function chooses this proposal, adding it to the $History$ and Ag_1 is ready to start the negotiation with the following proposal:
   `I propose that you provide me` *nail*`, because if I use` *hammer* `and` *nail* $\wedge$ *hammer* $\rightarrow$ *hangPicture*`, then I can achieve` *hangPicture* `in exchange for` *screw*`.`

2. Ag_2 receives Ag_1 proposal, and invokes the $Answer_2$ algorithm. Ag_2 adds the proposal to his $History$ and then uses the Gen_2 function to compute $proposal\mathcal{S}_2$.

$$\begin{aligned} prop\mathcal{S}_2 &= \mathsf{Gen}_2(R_2 \cup P_2, \emptyset, \{hangMirror\}) \\ &= \{(\{hammer\}, R_2 \cup P_2, \{hangMirror\}, \emptyset)\} \end{aligned}$$

Since $\odot prop \notin \bigodot prop\mathcal{S}_2$ (i.e. $(\{screw\},\{nail\}) \notin \{(\{hammer\},\emptyset)\})$ Ag_2 can use the proposal information to update his beliefs, and his $Init_2$ function to generate a proposal to answer Ag_1. The current mental state of Ag_2 is now as follows:

$$\begin{array}{ll} History_2 = [\,] & P_2 = \{hammer \wedge nail \rightarrow hangMirror\} \\ R_2 = \{nail\} & BR_2 = \{nail \wedge hammer \rightarrow hangPicture\} \\ G_2 = \{hangMirror\} & BG_2 = \{hangPicture\} \\ & BP_2 = \{\} \end{array}$$

The whole dialogue obtained in the negotiation program for this scenario is the following:

```
1 Says: I propose that you provide me [nail] because if I use
  [hammer, nail&hammer=>hangPicture] then I can achieve
  [hangPicture] in exchange for [screw]
2 Says: I propose that you provide me [hangMirror] because if I
  use [] then I can achieve [hangMirror] in exchange for [nail]
1 Says: I propose that you provide me [nail] because if I use
  [hammer, nail&hammer=>hangPicture] then I can achieve [hangPicture]
  in exchange for [screw, screwDriver, screwDriver&screw=>hangMirror]
2 Says: Accept, I give you [nail] and
  you give me [screw, screwDriver, screwDriver&screw=>hangMirror]
```

5 Conclusions

In this paper we have presented a novel approach to automated negotiation between two agents based on belief revision operators. In order to achieve their goals, agents engage in a cooperative dialogue, exchanging information about which resources and plans they have, and the possible exchanges they are willing to carry out. Our approach to automated negotiation was formalized in COQ, and the associated algorithms were implemented in logic programming. A revised version of the HNM was solved, showing how the agents can negotiate starting with incomplete (or wrong) information about the other agent's beliefs.

It must be noted that there have been other approaches to integrating belief revision and negotiation. In [10] a formal characterization of negotiation from a belief revision perspective is given, but no implementation issues are considered. In contrast with the original argumentative framework to solve the HNM problem in [6], our negotiation model allows the agents to gain and revise their beliefs as the dialogue takes place. Consequently, in our approach an agent does not need to have initial beliefs about the other agent involved in the negotiation. In [7] a similar scenario is analyzed, but agents are aware of all the agents' resources and the agents' plans are not negotiable. We think that our proposal is more flexible in this respect, as plans are also negotiation objects in our formalization.

Formal models of belief change can be very helpful in providing suitable frameworks for rational agents [2], in which the information from interagent dialogues can be better exploited. Part of our future work is focused on studying

complexity issues related with our proposal, as done by Zhang in the context of belief-revision based bargaining and negotiation [9]. Furthermore, we are interested in extend our approach to a multiagent platform considering also the possibility that different agents may have different languages following the proposal of Son et al. [8]. We are also investigating the logical properties of our approach, particularly those concerning the chracterization of different incision and selection functions. Research in these directions is currently being pursued.

References

1. C. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Log.*, 50(2):510–530, 1985.
2. G. Bonanno, J. Delgrande, J. Lang, and H. Rott. Special issue on formal models of belief change in rational agents. *J. Applied Logic*, 7(4):363, 2009.
3. E. Fermé, K. Saez, and P. Sanz. Multiple kernel contraction. *Studia Logica: An International Journal for Symbolic Logic*, 73(2):pp. 183–195, 2003.
4. S. Hansson. Kernel contraction. *The Journal of Symbolic Logic*, 59(3):pp. 845–859, 1994.
5. N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: Prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.
6. S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.
7. I. Rahwan, P. Pasquier, L. Sonenberg, and F. Dignum. On the benefits of exploiting underlying goals in argument-based negotiation. In *Twenty-Second Conference on Artificial Intelligence (AAAI)*, pages 116–121, Vancouver, 2007.
8. T. Cao Son, E. Pontelli, and C. Sakama. Logic programming for multiagent planning with negotiation. In *Proc. of the 25th Intl. Conference on Logic Programming (LNCS 5649)*, pages 99–114. Springer, 2009.
9. D. Zhang. A logic-based axiomatic model of bargaining. *Artif. Intell.*, 174(16-17):1307–1322, 2010.
10. D. Zhang, N. Foo, T. Meyer, and R. Kwok. Negotiation as mutual belief revision. In *In Proceedings of AAAI04*, pages 317–322, 2004.