

Negotiation Among DDeLP Agents

Fernando A. Tohmé¹ and Guillermo R. Simari²

¹ Department of Economics and National Research Council (CONICET)

EMAIL: ftohme@criba.edu.ar

² Artificial Intelligence Research and Development Laboratory

Department of Computer Science and Engineering

EMAIL: grs@cs.uns.edu.ar

UNIVERSIDAD NACIONAL DEL SUR

Baha Blanca – ARGENTINA

Abstract. Negotiation can be conceived as the exchange of messages among self-interested agents in order to settle on an agreement over a given issue. They decide which messages to send according to their preferences and their evolving beliefs. Agents able to handle this dynamics of messages and beliefs can be represented by means of Defeasible Logic Programming augmented with utility functions. This approach to argumentation has the advantage of providing a useful platform for the representation of beliefs and the generation of messages. The interactive nature of negotiations requires an updating mechanism to be applied over the knowledge bases of the agents. The features of this mechanism are described by a *protocol* of a negotiation. Although there are many possible protocols, we concentrate on one that ensures the existence of an agreement in negotiations. The formalism of DeLP provides a very natural approach to the characterization of such a protocol.

1 Introduction and Motivation

The design of agents able to engage in negotiations is one of the main goals in the research on Multi-Agent Systems [LS01]. Justifications for the behavior of agents in negotiations have been known for a long time in disciplines related to the study of Decision-Making processes [Mye89]. Despite the efforts of many authors both in the Decision Sciences as in Multi-Agents Systems, the characterization of precise mechanisms of interaction in negotiations has shown to be a hard problem because, unlike markets and voting situations, the context of interaction varies from a negotiation to another. An abstract characterization of the conditions for mechanisms (*protocols*) that may ensure the convergence to agreements in negotiations has been presented in [Toh02]. We take up some of the ideas there, but adapted to the additional requirement of giving a precise foundation for the internal argumentation processes carried out by each individual agent.

To represent those deliberation processes we choose an alternative form of declarative programming, Defeasible Logic Programming (DeLP) [SCG94, Gar00, GS04]. This formalism combines Logic Programming

with Defeasible Argumentation [Pol87,SL92,Pol95,Dun95,CML00,PV00], allowing the representation of tentative knowledge and leaving for the inference mechanism the task of finding the conclusions that the knowledge base warrants [CDSS03]. Furthermore, we introduce Decision-Theoretic tools into DeLP, in order to represent the fact that agents are self-interested. This is achieved by adding preferences to the formalism of DeLP, *i.e.* *utility* considerations [Lou90,TS04]. We call agents that reason using this Decision-theoretic enhanced Defeasible Logic Programming, *DDeLP agents*.

We consider only two-agent negotiations, since the extension to any number of agents follows basically the same pattern but is more involved in its syntax, without providing extra intuitions. In this simple kind of negotiations, one agent (by convention she is always called \mathbf{a}_1), chooses one preferred conclusion derived from her beliefs. This conclusion represents a possible settlement for the negotiation. The other agent (\mathbf{a}_2) treats this message as a query Q . Agent \mathbf{a}_2 *agrees* with \mathbf{a}_1 if he can find a warranted argument \mathcal{A} for Q . Otherwise, he explores his own knowledge base in order to select a new proposal to make.

If we want to ensure the convergence of the negotiation, we may impose over the agents a protocol to guide the process of exchange of messages towards an agreement [Lou98]. As shown in [Toh02] a sufficient condition for such a protocol is its *monotonicity*. One consequence of this property is the absence of cycles, understood as the repetition of messages. Since a message cannot be repeated by either of the agents, any attempt to “convince” the other party in the negotiation is implicitly ruled out. In fact, a monotonic protocol rather forces an agent to accept the message of the other as a constraint.³ Therefore, returning to our Decision-theoretic DeLP framework, if no warranted argument for the message of the other party is found, the agent must look for at least one rule responsible for this. From among several, she should choose the one that yields less utility and eliminate it from the knowledge base. Although there are other possible responses, this procedure represents a very cautious change of beliefs of the agent. With the corrected knowledge base, which represents the current beliefs of the agent, a message is chosen and the process repeats itself until either a query becomes warranted or there are no longer rules to be deleted from the knowledge base. In the first case the negotiation is said to end in an agreement. In the latter case, instead, it results in the breakup of the negotiation (which can be seen as a form of degenerate agreement on not pursuing further the negotiation).

The plan of the rest of this paper is as follows. In section 2 we will present the rudiments of DeLP with utilities. In section 3 we introduce the protocol and describe how it proceeds. Section 4 discusses possible extensions for this work.

2 Decision-theoretic Defeasible Logic Programming

We consider a language with three disjoint components:

³ This is a requirement shared by all known models of bargaining.

- *Facts*, which are ground literals representing atomic information (or the negation of atomic information).
- *Strict Rules* of the form $L_0 \leftarrow L_1, \dots, L_n$, where L_0 is the *head* and $\{L_i\}_{i>0}$ is the *body*. Each L_i in the body or the head is a literal.
- *Defeasible Rules* of the form $L_0 \prec L_1, \dots, L_n$, where L_0 is the *head* and $\{L_i\}_{i>0}$ is the *body*. Each L_i in the body or the head is a literal.

Then, a *Defeasible Logic Program* is a set of facts, strict rules, and defeasible rules. $\mathcal{P} = (\Pi, \Delta)$, where Π denotes the set of facts and strict rules, while Δ denotes the set of defeasible rules. For each query Q there are four possible answers: YES, NO, UNDECIDED or UNKNOWN.

To determine which answer is correct, we need the notion of *argument*. Given a program $\mathcal{P} = (\Pi, \Delta)$ and a literal L , $\langle \mathcal{A}, L \rangle$ is an argument structure for L . \mathcal{A} is a set of defeasible rules in Δ such that:

1. there exists a *defeasible derivation* of L from $\Pi \cup \mathcal{A}$. That is, there exists a finite sequence $L_1, \dots, L_n = h$ of ground literals, such that each L_i is either a fact in Π or there exists a rule in $\Pi \cup \mathcal{A}$ with L_i as its head, and every literal in the body B_j is such that $B_j \in \{L_k\}_{k<i}$,
2. there is no literal P such that both P and $\neg P$ have defeasible derivations from $\Pi \cup \mathcal{A}$,
3. \mathcal{A} is minimal, *i.e.*, there does not exist $\mathcal{A}_1 \subseteq \mathcal{A}$ such that \mathcal{A}_1 satisfies (1) and (2).

This framework can be enhanced by means of *preferences*, $\Phi : \Pi \cup \Delta \rightarrow \mathbf{B}$, where \mathbf{B} is an arbitrary Boolean algebra with top \top and bottom \perp . The new elements $\Phi(\cdot)$ and \mathbf{B} represent explicit preferences, in the sense that given two pieces of information $\mu_1, \mu_2 \in \Pi \cup \Delta$ if μ_1 is strictly *more preferred* than μ_2 then $\Phi(\mu_1) \succ_{\mathbf{B}} \Phi(\mu_2)$, where $\succeq_{\mathbf{B}}$ is the order of \mathbf{B} . The elements of $\mu \in \Pi \cup \Delta$ which are most preferred receive a label $\Phi(\mu) = \top$.

We do not assume here that Φ assigns \top to all strict rules in Π , and not even that $\Phi(\mu_1) \succ_{\mathbf{B}} \Phi(\mu_2)$ for $\mu_1 \in \Pi$ and $\mu_2 \in \Delta$. This is because $\Phi(\cdot)$ has, unlike the distinction between *strict* and *defeasible* rules, no epistemic content. Instead, the preferences represent other kinds of rationales. In particular the cost-benefit rates of the pieces of information (since their use may preclude the use of other pieces in the reasoning process).

Whatever the reasons are for preferring elements of $\Pi \cup \Delta$, we postulate a Boolean algebra \mathbf{B} over which $\Phi(\cdot)$ ranges. It can be argued that a more general ordering could be appropriate but, as we will see, the inference engine has to perform some operations over the labels of the pieces of information used in the process of argumentation. In consequence, the range of $\Phi(\cdot)$ has to be not only an ordered set but also be closed under the logic operators \bigwedge and \bigvee . This can be easily represented by means of a Boolean algebra. In the simplest case, in which \mathbf{B} is just a compact subset of real numbers with the natural order, we may say that $\Phi(\mu)$ is the *utility* of the piece of information μ .

From the preferences over $\Pi \cup \Delta$, we can find *preferential values* over defeasible derivations. A fact L , which can be seen as the head of a (strict) rule

with an empty body, has a value denoted $V(L, \emptyset) = \Phi(L)$. By induction, given rule μ (strict or defeasible) with head L and body B_1, \dots, B_m , if L is derived using μ its preferential value is $V(L, \mu) = \Phi(\mu) \wedge \bigwedge_{k=1}^m V(B_k)$. The intuition here is that a conclusion is as strongly preferred as the weakest of either its premises or the rule used in the derivation. Given a defeasible derivation from $\Pi \cup \Delta$, $\mathcal{L}_{\Pi \cup \Delta}(L) : L_1, \dots, L_n = h$, it yields for its conclusion L a value $V(h, \mathcal{L}_{\Pi \cup \Delta}(L)) = V(L, \mu)$ where μ is the rule that yields $h = L_n$ up from some literals in $\{L_j\}_{j < n}$.

By extension, an argument structure $\langle \mathcal{A}, L \rangle$ yields a value for L , $V(L, \mathcal{A}) = \bigwedge_{\mathcal{L}_{\Pi \cup \mathcal{A}}(L)} V(L, \mathcal{L}_{\Pi \cup \mathcal{A}}(L))$. That is, it yields the lowest value among all the derivations of L by using defeasible rules in \mathcal{A} . Notice that, by definition of \mathcal{A} there is no other set $\mathcal{A}' \subset \mathcal{A}$ that allows the derivation of L , but more than one selection of strict rules may exist in Π that allows, jointly with \mathcal{A} , to do that.

Let \mathcal{F} the set of all literals that can have a defeasible derivation from $\Pi \cup \Delta$. Any subset $H \subseteq \mathcal{F}$ has a value $V(H) = \bigvee_{L \in H} \bigwedge_{\mathcal{L}_{\Pi \cup \Delta}(L)} V(L, \mathcal{L}_{\Pi \cup \Delta}(L))$. This means that H is as valuable as the most valuable of its elements, which in turn is as valuable as the weakest of its derivations.

With this characterization we speak of an Decision-theoretic enhanced Defeasible Logic Program or $\mathcal{P}' = (\Pi, \Delta, \Phi, \mathbf{B})$ which is intended to provide answers to queries through a process of argumentation that proceeds making comparisons among arguments. The main criterion of comparison used is *preferential specificity* [Poo85,SL92,SGCS03,TS04]. Consider a program $\mathcal{P}' = (\Pi, \Delta, \Phi, \mathbf{B})$ with Π_G the set of strict rules from Π . Let \mathcal{F} the set of all literals that can have a defeasible derivation from $\Pi \cup \Delta$. Let $\langle \mathcal{A}_1, L_1 \rangle$ and $\langle \mathcal{A}_2, L_2 \rangle$ be two argument structures with $L_1, L_2 \in \mathcal{F}$. Then $\langle \mathcal{A}_1, L_1 \rangle$ is *strictly more preferentially specific than* $\langle \mathcal{A}_2, L_2 \rangle$ if:

1. For all $H \subseteq \mathcal{F}$, if there exists a defeasible derivation of L_1 from $\Pi_G \cup H \cup \mathcal{A}_1$ while $\Pi_G \cup H \not\vdash L_1$, then L_2 can be defeasibly derived from $\Pi_G \cup H \cup \mathcal{A}_2$, and
2. there exists $H' \subseteq \mathcal{F}$ such that there exists a defeasible derivation of L_2 from $\Pi_G \cup H' \cup \mathcal{A}_2$ and $\Pi_G \cup H' \not\vdash L_2$ but there is no defeasible derivation of L_1 from $\Pi_G \cup H' \cup \mathcal{A}_1$.
3. For every H verifying (1) and H' verifying (2), $V(H) \succeq_{\mathbf{B}} V(H')$.

Argument $\langle \mathcal{A}_1, L_1 \rangle$ *counterargues* another $\langle \mathcal{A}_2, L_2 \rangle$ at a literal L if there exists a sub-argument of $\langle \mathcal{A}_2, L_2 \rangle$, $\langle \mathcal{A}, L \rangle$, *i.e.*, $\mathcal{A} \subseteq \mathcal{A}_2$, such that there exists a literal P verifying both $\Pi \cup \{L, L_1\} \vdash P$ and $\Pi \cup \{L, L_1\} \vdash \neg P$.

If $\langle \mathcal{A}_1, L_1 \rangle$ and $\langle \mathcal{A}_2, L_2 \rangle$ are two argument structures, $\langle \mathcal{A}_1, L_1 \rangle$ is a *proper preferential defeater* for $\langle \mathcal{A}_2, L_2 \rangle$ at literal L iff there exists a sub-argument of $\langle \mathcal{A}_2, L_2 \rangle$, $\langle \mathcal{A}, L \rangle$ such that $\langle \mathcal{A}_1, L_1 \rangle$ counterargues $\langle \mathcal{A}_2, L_2 \rangle$ at L and $\langle \mathcal{A}_1, L_1 \rangle$ is strictly more preferentially specific than $\langle \mathcal{A}, L \rangle$. Alternatively, $\langle \mathcal{A}_1, L_1 \rangle$ is a *blocking preferential defeater* for $\langle \mathcal{A}_2, L_2 \rangle$ at literal L iff there exists a sub-argument of $\langle \mathcal{A}_2, L_2 \rangle$, $\langle \mathcal{A}, L \rangle$ such that $\langle \mathcal{A}_1, L_1 \rangle$ counterargues $\langle \mathcal{A}_2, L_2 \rangle$ at L and neither $\langle \mathcal{A}_1, L_1 \rangle$ is strictly more preferentially specific than $\langle \mathcal{A}, L \rangle$ nor is $\langle \mathcal{A}, L \rangle$ strictly

more preferentially specific than $\langle \mathcal{A}, L \rangle$. If $\langle \mathcal{A}_1, L_1 \rangle$ is either a proper or a blocking preferential defeater of $\langle \mathcal{A}_2, L_2 \rangle$, it is said to be a *preferential defeater* of the latter.

An *argumentation line* for an argument structure $\langle \mathcal{A}_0, L_0 \rangle$ is a sequence $\Gamma = [\langle \mathcal{A}_0, L_0 \rangle, \langle \mathcal{A}_1, L_1 \rangle, \langle \mathcal{A}_2, L_2 \rangle, \dots]$ where for each $i > 0$ $\langle \mathcal{A}_{i+1}, L_{i+1} \rangle$ is a defeater of $\langle \mathcal{A}_i, L_i \rangle$. $\Gamma_S = [\langle \mathcal{A}_0, L_0 \rangle, \langle \mathcal{A}_2, L_2 \rangle, \langle \mathcal{A}_4, L_4 \rangle, \dots]$ is the sequence of *supporting* argument structures of Γ , while the sequence of *interfering* ones is $\Gamma_I = [\langle \mathcal{A}_1, L_1 \rangle, \langle \mathcal{A}_3, L_3 \rangle, \langle \mathcal{A}_5, L_5 \rangle, \dots]$.

An *acceptable* argumentation line in a defeasible program $\mathcal{P}' = (\Pi, \Delta, \Phi, \mathbf{B})$ is a finite sequence $\Gamma = [\langle \mathcal{A}_0, L_0 \rangle, \dots, \langle \mathcal{A}_n, L_n \rangle]$ such that:

1. Both Γ_S and Γ_I are *concordant*, i.e., there is no P such that both P and $\neg P$ have defeasible derivations from $\Pi \cup \bigcup_{i=0}^{\lfloor \frac{n}{2} \rfloor} \mathcal{A}_{2i}$ and no P' with defeasible derivations for both P' and $\neg P'$ from $\Pi \cup \bigcup_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} \mathcal{A}_{2i+1}$.
2. No argument $\langle \mathcal{A}_k, L_k \rangle \in \Gamma$ is a subargument of an argument $\langle \mathcal{A}_j, L_j \rangle$, i.e., $\mathcal{A}_k \not\subseteq \mathcal{A}_j$, for $j < k$.
3. For each $i < n$, if $\langle \mathcal{A}_i, L_i \rangle$ is a blocking preferential defeater of $\langle \mathcal{A}_{i-1}, L_{i-1} \rangle$ then $\langle \mathcal{A}_{i+1}, L_{i+1} \rangle$ is a proper preferential defeater of $\langle \mathcal{A}_i, L_i \rangle$.

To answer a query Q , the *preferential warrant procedure* builds up a candidate argument structure $\langle \mathcal{A}, Q \rangle$. Then, it associates to this argument a *preferential dialectical tree* $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}$ as follows:

1. The root of the tree is labeled, $\langle \mathcal{A}_0, Q_0 \rangle$, i.e., $\mathcal{A}_0 = \mathcal{A}$ and $Q_0 = Q$.
2. Let n be a non-root node, with label $\langle \mathcal{A}_n, Q_n \rangle$ and $\Gamma = [\langle \mathcal{A}_0, Q_0 \rangle, \dots, \langle \mathcal{A}_n, Q_n \rangle]$ the labels in the path from the root to n . Let $\mathcal{B} = \{\langle \mathcal{B}_1, H_1 \rangle, \dots, \langle \mathcal{B}_k, H_k \rangle\}$ be the set of all the preferential defeaters for $\langle \mathcal{A}_n, Q_n \rangle$. For $1 \leq i \leq k$, if $\Gamma' = [\langle \mathcal{A}_0, Q_0 \rangle, \dots, \langle \mathcal{A}_n, Q_n \rangle, \langle \mathcal{B}_i, H_i \rangle]$ is an acceptable argumentation line, n has a child n_i labeled $\langle \mathcal{B}_i, H_i \rangle$. If $\mathcal{B} = \emptyset$ or no $\langle \mathcal{B}_i, H_i \rangle \in \mathcal{B}$ is such that Γ' is acceptable, then n is a leaf of the tree.

The nodes of $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}$ can be marked, yielding a tagged tree $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}^*$ as follows:

- All leaves of $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}$ are marked U in $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}^*$.
- If $\langle \mathcal{B}, H \rangle$ is the label of a node which is not a leaf, the node will be marked U in $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}^*$ if every child is marked D . Otherwise, if at least one of its children is marked U , it is marked as D .

Then, given an argument $\langle \mathcal{A}, Q \rangle$ and its associated tagged tree $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}^*$, if the root is marked U , the literal Q is said to be *preferentially warranted*. \mathcal{A} is said to be the *preferential warrant* for Q . Therefore, given a query Q the possible answers will be:

- | | |
|------------|---|
| YES, | if Q is preferentially warranted; |
| NO, | if $\neg Q$ is preferentially warranted; |
| UNDECIDED, | if neither Q nor $\neg Q$ are preferentially warranted; |
| UNKNOWN, | if Q is not in the language of the program. |

3 Negotiation

A negotiation can be seen as the exchange of messages among agents in order to reach an agreement over a given issue. The main elements in a negotiation are the following:

- The possible settlements.
- The preferences over them.
- The individual beliefs about the possible results.
- The messages that can be exchanged.

The messages are chosen according to the preferences over the class of settlements that are believed to be acceptable. Once a message is received, an agent has to decide whether to accept the implied settlement or to break up the negotiation or to explore for new possibilities. If the latter is the case, the agent has to update her beliefs and choose her message according to that.

In terms of DDeLP consider a program $\mathcal{P}' = (\Pi, \Delta, \Phi, \mathbf{B})$ where $\Phi = \Phi_1 \times \dots \times \Phi_n$ with range \mathbf{B}^n , from which we may define the elements of a n -person negotiation as follows:

- The possible settlements are the literals L that can be defeasibly derived from $\Pi \cup \Delta$.
- The preferences over the literals are derived from Φ .
- The individual beliefs are subsets of $\Pi \cup \Delta$.
- The messages are the literals L plus two extra symbols, “*yes*” and “*break*”, to indicate either agreement or the breakup of negotiations.

That is, we assume that agents consider only a certain subset of rules, from which they select some literals as both possible settlements and as messages. We consider only two agents, 1 and 2. Each agent i performs, at each stage t of the negotiation, a DDeLP program $\langle \Pi_i^t, \Delta_i^t, \Phi_i^t, \mathbf{B} \rangle$, where $\Pi_i^t \cup \Delta_i^t \subseteq \Pi \cup \Delta$ and Φ_i^t is the restriction of Φ_i over $\Pi_i^t \cup \Delta_i^t$.

At $t = 1$ agent \mathbf{a}_1 sends a message (a literal L^1 derived from $\Pi_1^1 \cup \Delta_1^1$) to agent \mathbf{a}_2 . He may query his program $\Pi_2^2 \cup \Delta_2^2$ and if the answer is YES end the negotiation by accepting L and sending the message “*yes*”. Otherwise, if the answer is UNKNOWN he may break the negotiation (because the issue has become meaningless) and send “*break*”. In case that the answer is either NO or UNDECIDED, a new literal is chosen to be send as a message.

In general, a response to a message L^t received by agent \mathbf{a}_i at round $t + 1$ of the negotiation, is a message L^{t+1} defeasibly derived from $\Pi_i^{t+1} \cup \Delta_i^{t+1}$. The exchange of messages proceeds in orderly fashion: agent \mathbf{a}_1 sends her messages at odd values of t (*i.e.* at 1, 3, ...) while \mathbf{a}_2 sends hers at even values (2, 4, ...).

A possibility is that L^t is either NO or UNDECIDED. Then, $\Pi_i^{t+1} \cup \Delta_i^{t+1}$ must be revised and updated according to L^t . The result of this operation of updating is $\bar{\Pi}_i^{t+1} \cup \bar{\Delta}_i^{t+1}$ ⁴. There are alternative characterizations of this updating opera-

⁴ The next time agent i receives a message, $t + 3$, his knowledge base will be $\Pi_i^{t+3} \cup \Delta_i^{t+3} \equiv \bar{\Pi}_i^{t+1} \cup \bar{\Delta}_i^{t+1}$.

tion [FKIS02], but a mandatory requirement is that it must be consistent with the protocol of negotiation.

Since we assume that the goal of both agents is either to reach an agreement or break up the negotiation, any sufficient condition that ensures such result may be applied to define a protocol. In [Toh02] it is shown that such condition is *monotonicity*, in the sense of reducing disagreements. In other words, agents are allowed to exchange messages (without repetition) until either an agreement is found or the negotiation breaks up.

In this sense, if until stage t the messages that have been sent back and forth are L^1, L^2, \dots, L^{t-1} , the possible messages to be send from then on are constrained by the protocol.

The following is obviously a monotonic protocol:

Protocol 31 *If an agreement nor a break up is reached, the response to a message L^t must be a L^{t+1} in $\mathcal{L} \setminus \{L^j\}_{j=1}^t$, where \mathcal{L} is the set of literals that can be defeasibly derived from $\Pi \cup \Delta$.*

If such non-deterministic protocol is applied, a process of updating $\Pi_i^{t+1} \cup \Delta_i^{t+1}$ must be consistent with it. In this sense, if L^t is not preferentially warranted, no L^j for $j = 1, \dots, t$ should be used as a message up from the resulting knowledge base $\bar{\Pi}_i^{t+1} \cup \bar{\Delta}_i^{t+1}$.

This means that, in particular, the current beliefs must be changed. The beliefs should no longer allow the messages sent in the previous rounds to be considered warranted.⁵ One way to achieve this is by using the following procedure:

Procedure 31 *Consider, for a given i , the tagged trees \mathcal{T}_i^k , for $k = 1, \dots, t$. Among those with roots marked U , choose the leaves $\langle \mathcal{H}^k, H^k \rangle$ that minimize $V_i(H^k, \mathcal{H}^k)$, derived from Φ_i^{t+1} .*

For each of those leaves, choose a rule that minimizes $\Phi_i^{t+1}(\mu^k)$ over all the rules that participate in the derivation of H^k from $\Pi^{t+1} \cup \mathcal{H}^k$. Then define $\bar{\Pi}_i^{t+1} \cup \bar{\Delta}_i^{t+1} := (\Pi_i^{t+1} \cup \Delta_i^{t+1}) \setminus \{\mu\}_k$.

This procedure, used in the following algorithm allows to find the updated beliefs and choose next message:

Algorithm 31 [Update Beliefs and Select Message]

1. Define $\mathcal{T}_{\langle \mathcal{A}, L^k \rangle}$ for $k = 1, \dots, t$.
2. Run Procedure 3.1.
3. Find $\{\mathcal{T}_i^k\}_{k=1}^t$ over $\bar{\Pi}_i^{t+1} \cup \bar{\Delta}_i^{t+1}$
4. If a root is marked U go to 2. Else
 - (a) If $\bar{\Pi}_i^{t+1} \cup \bar{\Delta}_i^{t+1} = \emptyset$, send the message “break”.

⁵ As said, preferences could also change, but this equivalent to replace Φ_i for Φ'_i . Although this can be easily introduced in our framework, we leave the details for an extension of this work.

(b) Else choose L^{t+1} such that minimizes $\Phi_i^{t+1}(L)$ over those literals L that can be preferentially warranted in $\bar{\Pi}_i^{t+1} \cup \bar{\Delta}_i^{t+1}$.

It is immediate that:

Proposition 31 *Algorithm 3.1 implements Protocol 3.1.*

Proof: *Trivial. If L^t is warranted, then an agreement is reached and the selected message is “yes”, else if L^t is UNKNOWN the negotiation breaks up. Otherwise, Algorithm 3.1 is such that if $\bar{\Pi}_i^{t+1} \cup \bar{\Delta}_i^{t+1} = \emptyset$ the negotiation breaks up, otherwise, it ensures that L^1, \dots, L^t are not preferentially warranted in $\bar{\Pi}_i^{t+1} \cup \bar{\Delta}_i^{t+1}$, therefore it chooses $L^{t+1} \in \mathcal{L} \setminus \{L^j\}_{j=1}^t$.*

Notice that Algorithm 3.1 is not the only possible implementation of Protocol 3.1, since the latter just asks for monotonicity in the messages, while the Algorithm intends to find the *best* messages for agent \mathbf{a}_i .

Another result that follows is:

Proposition 32 *If both agents use Algorithm 3.1 to choose messages, the negotiation either ends in an agreement or in a break up.*

Proof: *Algorithm 3.1 implements Protocol 3.1. We denote $\mathcal{L}^t = \mathcal{L} \setminus \{L^k\}_{k=1}^t$. There are two possibilities, either there exists a T such that when one of the agents sends a message L^T the other agent responds with “yes”. Alternatively, if there is no L that may result in an agreement, as a consequence of the Compactness Theorem for First-Order Logic there exists a $T' < \infty$ such that $\mathcal{L}^{T'} = \emptyset$ and leads to a response “break”.*

A final consequence of using Algorithm 3.1 is that if agreements (including break ups as degenerate agreements) are *path dependent*. That is, the choices made by the agents condition further choices. Therefore, over the same knowledge base, the agents may end up agreeing on different conclusions. The following example shows this:

Example 31 *Consider the classical example in defeasible argumentation where preferences are defined for $\mathbf{B} = \{0, 1\}$, with $0 < 1$. The preferences, which for simplicity are assumed common to both agents, are indicated in parentheses next to the corresponding pieces of information:*

$$\begin{aligned} \Pi &= \{bird(X) \prec penguin(X) (1), penguin(tweety) (1), bird(tweety) (1)\} \\ \Delta &= \{\neg flies(X) \prec penguin(X) (1), flies(X) \prec bird(X) (0.5)\} \end{aligned}$$

Agents have different beliefs:

$$\begin{aligned} \Pi_1^1 &= \{penguin(tweety) (1), bird(tweety) (1)\} \\ \Delta_1^1 &= \{\neg flies(X) \prec penguin(X) (1)\} \end{aligned}$$

while

$$\begin{aligned}\Pi_2^1 &= \{bird(X) \multimap penguin(X) (1), penguin(tweety) (1), \} \\ \Delta_2^1 &= \{flies(X) \multimap bird(X) (0.5)\}\end{aligned}$$

For agent \mathbf{a}_1 , there are two warranted conclusions, $penguin(tweety)$ and $\neg flies(tweety)$. Suppose that her message is $L^1 = penguin(tweety)$. Since it is also a warranted conclusion for \mathbf{a}_2 , he will respond with “yes”. Otherwise, suppose that $L^1 = \neg flies(tweety)$. Since for \mathbf{a}_1 $flies(tweety)$ is warranted he must apply Algorithm 3.1. It follows that he has to drop the rule with lowest preference, namely $flies(X) \multimap bird(X)$. Then,

$$\begin{aligned}\Pi_2^2 &= \{bird(X) \multimap penguin(X) (1), penguin(tweety) (1), \} \\ \Delta_2^2 &= \emptyset\end{aligned}$$

The only (trivially) warranted conclusions are $penguin(tweety)$ and $bird(tweety)$. In the case that $L^2 = penguin(tweety)$, the response is “yes” and the same is true if $L^2 = bird(tweety)$.

On the other hand, assume that $\Phi(flies(X) \multimap bird(X)) = 1$ while $\Phi(bird(X) \multimap penguin(X)) = 0.5$. Accordingly, if $L^1 = \neg flies(tweety)$ then

$$\begin{aligned}\Pi_2^2 &= \{ penguin(tweety) (1), \} \\ \Delta_2^2 &= \{flies(X) \multimap bird(X) (1)\}\end{aligned}$$

Therefore, $L^2 = penguin(tweety)$, and the response is “yes”.

Finally, if

$$\begin{aligned}\Pi_2^1 &= \{bird(tweety) (1)\} \\ \Delta_2^1 &= \{flies(X) \multimap bird(X) (0.5)\}\end{aligned}$$

and $L^1 = penguin(tweety)$, the response is “break”.

4 Conclusions

We presented in this paper a framework of negotiation with DeLP extended with preferences. An algorithm of belief updating based on the elimination of rules ensures that agents will reach an agreement, although this agreement depends on the particular sequence of messages chosen.

A matter of further work will be to see if a negotiation still converges to an agreement if another mechanism of belief updating is used.

5 Acknowledgments

This research was partially supported by CONICET, by the Secretaría General de Ciencia y Tecnología de la Universidad Nacional del Sur and by Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 No. 13096). The authors would like to thank anonymous reviewers for providing helpful comments to improve the final version of this paper.

References

- [CDSS03] Carlos I. Chesñevar, Jürgen Dix, Frieder Stolzenburg, and Guillermo R. Simari. Relating defeasible and normal logic programming through transformation properties. *Theoretical Computer Science*, 290(1):499–529, Jan 2003.
- [CML00] Carlos I. Chesñevar, Ana G. Maguitman, and Ronald P. Loui. Logical Models of Argument. *ACM Computing Surveys*, 32(4), December 2000.
- [Dun95] Phan M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming and n -person games. *Artificial Intelligence*, 77:321–357, 1995.
- [FKIS02] Marcelo A. Falappa, Gabrielle Kern-Isberner, and Guillermo R. Simari. Explanations, belief revision and defeasible reasoning. *Artificial Intelligence Journal*, 141(1-2):1–28, October 2002.
- [Gar00] Alejandro J. García. *Defeasible Logic Programming: Definition, Operational Semantics and Parallelism*. PhD thesis, Computer Science and Engineering Department, Universidad Nacional del Sur, Bahía Blanca, Argentina, December 2000.
- [GS04] Alejandro J. García and Guillermo R. Simari. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*, 4(1):95–138, 2004.
- [KSE98] Sarit Kraus, Katia Sycara, and Amir Evenchik. Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence*, 104(1–2):1–69, 1998.
- [Lou90] Ronald P. Loui. Defeasible specification of utilities. In Henry Kyburg, Ronald Loui, and Greg Carlson, editors, *Knowledge Representation and Defeasible Reasoning*, pages 345–359. Kluwer Academic Publishers, Dordrecht, 1990.
- [Lou98] Ronald P. Loui. Process and policy: Resource-bounded nondemonstrative reasoning. *Computational Intelligence: An International Journal*, 14, 1998.
- [LS01] Kate Larson and Tuomas Sandholm. Bargaining with limited computation: Deliberation equilibrium. *Artificial Intelligence*, 132, 2001.
- [Mye89] Roger B. Myerson. Credible negotiation statements and coherent plans. *Journal of Economic Theory*, 48, 1989.
- [Pol87] John Pollock. Defeasible Reasoning. *Cognitive Science*, 11:481–518, 1987.
- [Pol95] John Pollock. *Cognitive Carpentry: A Blueprint for How to Build a Person*. MIT Press, 1995.
- [Poo85] David L. Poole. On the Comparison of Theories: Preferring the Most Specific Explanation. In *Proc. 9th IJCAI*, pages 144–147. IJCAI, 1985.
- [PV00] Henry Prakken and Gerard Vreeswijk. Logical systems for defeasible argumentation. In D.Gabbay, editor, *Handbook of Philosophical Logic, 2nd ed.* Kluwer Academic Pub., 2000.

- [SCG94] Guillermo R. Simari, Carlos I. Chesñevar, and Alejandro J. García. The role of dialectics in defeasible argumentation. In *XIV International Conference of the Chilean Computer Science Society*, November 1994.
- [SGCS03] Frieder Stolzenburg, Alejandro J. García, Carlos I. Chesñevar, and Guillermo R. Simari. Computing generalized specificity. *Journal of Applied Non-Classical Logics*, 13(1):87–113, January 2003.
- [SL92] Guillermo R. Simari and Ronald P. Loui. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence*, 53:125–157, 1992.
- [Syc89] Katia Sycara. Multi-agent compromise via negotiation. In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence (Vol. 2)*. Morgan Kaufmann, Los Altos, CA, September 1989.
- [Syc90] Katia Sycara. Persuasive argumentation in negotiation. *Theory and Decision*, 28(3):203–242, May 1990.
- [Toh02] Fernando Tohmé. Negotiation and defeasible decision making. *Theory and Decision*, 53(4):289–311, 2002.
- [TS04] Fernando A. Tohmé and Guillermo R. Simari. Preferential defeasibility: Utility in defeasible logic programming. In *Proceedings of the 10th International Workshop on Non-Monotonic Reasoning*, pages 394–399. NMR, June 2004.