

# Pre-Argumentative Reasoning

Paulo Novais<sup>1</sup>, Luís Brito and José Neves

Departamento de Informática, Universidade do Minho, Braga , Portugal

## Abstract

Intelligent agents configure a new generation of virtual entities that perform various autonomous tasks on behalf of others, namely the humans. On the other hand, the information society requires the development of new and more intelligent methods, tools and theories to analyse, define, model and specify agent-based systems. It is under this presupposition that in this work are introduced a pre-argumentative reasoning scheme, which need to be able to take into account, in a negotiation, factors such as temporality, priority, delegation, gratitude and agreement, enabling the agents to react and pro-act accordingly. This argument-based negotiation among agents has much to gain from the use of Extended Logic Programming and Incomplete Information, in terms of the argument's evaluation. Indeed, it is based on this substratum that are presented the bases for a pre-contract negotiation via argumentation, where pre-contract negotiation will be defined as a protocol which, once enforced, will allow the exchange of messages containing proposals, counter-proposals, critiques, justifications or even explanations. The terms of a negotiation are also set, adjusting the contract according to the agent's knowledge base. In fact, an important contribution of this work relies on the presentation of the basic definitions and the general model of pre-contract based negotiations via argumentation.

**Keywords:** intelligent agents, logic, negotiation, argumentative systems.

## 1 Introduction

The amount of ambiguity present in real-life negotiations is intolerable for automatic reasoning systems. Formalizing the process that opposes, during negotiation, intelligent agents is extremely important. Logic, and especially, *Extend Logic Programming (ELP)* [2] poses itself as a powerful tool to achieve both the desired formality without compromising comprehension/readability, and the ability to easily build an executable prototype. Logical formulas are extremely powerful, unambiguous and possess a set of interesting advantages [8]:

*Expressing information in declarative sentences is far more modular than expressing it in segments of computer programs or in tables. Sentences can be true in a much wider context than specific programs can be used. The supplier of a fact does not have to*

*understand much about how the receiver functions or how or whether the receiver will use it. The same fact can be used for many purposes, because the logical consequences of collections of facts can be available.*

However, in a dynamic environment the simple use of logical formulas is not enough. The use of non-monotonic logics is mandatory; e.g., an agent may believe that something is true at a given point in time but it may conclude differently at a later time [10]. In *Extended Logic Programs (ELPs)*, queries may not be answered with a simple *true* or *false* statement. To represent incomplete information about the world, the possibility for the *unknown* exists, a situation that may be handled in terms of  $\neg$  (called *classical, strong* or *explicit* negation) and *not* (called *negation-by-failure*). In general logic programs, negative information is handled in terms of the *closed-world assumption* (i.e., everything that can not be proven to be *true* is taken as *false*). In *ELP* a query may fail due to the fact that *information is not available to support it* or, on the other hand, it may fail due to the fact that *negation succeeds*. The *Knowledge Base (KB)*, which serves as the basis for the agent's reasoning, can be seen has an *ELP program* in terms of a collection of rules in the form  $L_0 \leftarrow L_1, \dots, L_m, \text{not } L_{m+1}, \dots, \text{not } L_n$ , where  $L_i$  ( $0 \leq i \leq n$ ) is a *literal* (i.e., terms of the form  $p$  or  $\neg p$ , where  $p$  is an atom). This general form is reduced to  $L_0 \leftarrow$  (also represented as  $L_0$ ) in the case of factual information. The reasoning strategies behind each agent, take into consideration:

- **Architecture:** define and specify the agent's modules or functionalities and design the flow of information (e.g., *Beliefs, Desires and Intentions (BDI)* [14], *Experience-Based Mediator (EBM)* agents [11]);
- **Process quantification:** quantify each metric and/or sub-process with which the agents may have to deal with and establish the mechanisms and protocols for an efficient approach to a wide range of problems [3];
- **Reasoning mechanism:** each agent is in need of a formal set of statements that will serve as the main guidelines for the negotiation process; and
- **Process formalization:** the process of argumentation needs to proceed from a formal specification to a consistent implementation [4] in order to act/react in a reasonable (logic) way, accordingly to the circumstances. Arguing in such a context has many similarities to the process of legal arguing [13], where logic presents itself as the main specification and implementation tool.

The *negotiation* module present in the agent’s knowledge base, is the main object of the present work. In this module, the concepts of temporality, priority, delegation, gratitude and agreement are to be formally stated, which make the beliefs, desires and intentions around which an agent develops a negotiation act. Temporality introduces the temporal dimension on the base clauses (facts) that compose the *KB*, in terms of the non-destructive (non-monotonic) principles of knowledge assimilation. Priorities are paramount when reasoning takes the set of base clauses in order to justify/explain some previous action. Delegation is the underlying principle for the use of proxy agents. Gratitude expresses an active or passive form of mind that may condition, limit or restrict, the trade dealings among agents. Agreement is the principle undertaken among a set of agents in order to do something (e.g., to share some goal or knowledge), being the glue behind communitary action.

The main contributions of this work are: (i) the definition of a common ground to situate the agent’s reasoning mechanisms; (ii) the description of a reasoning mechanism necessary for a consistent and sound development of agents; (iii) the use of incomplete information in the reasoning process; and (iv) the establishment of sound syntactic and semantic tools for argument-based negotiation.

On section 2, the basic distinction between clauses expressing the notions of *capability* and *right to deal* is exposed. On section 3, the basic buildings blocks of the negotiation process such as theorem solvers, restrictions and null values are introduced, aiming at a proper formalization. On section 4, the process of reasoning with incomplete information is extended to include temporality and priorities, giving way to the formalization of concepts such as delegation, gratitude and agreement. On section 5, examples are presented. On section 6, related work is briefly analyzed and compared to the one hereby presented. Finally, on section 7, some conclusions are drawn.

## 2 Right to Deal – Basic Assumptions

During a negotiation process, each agent, although being able to deal with a counter-part, it may be inhibited to do so. Therefore, a distinction must be established between *capability* (i.e., an agent has the necessary expertise to do something) and *right* (i.e., an agent has the capability to do something and it can proceed that course of action) [9].

It is assumed that an agent has the ability to deal with every product, under any scenario. However, any agent has its behaviour conditioned by the *right to deal* premise. Two predicates will be considered:  $capability - to - deal : Product, Conditions, Counterpart \rightarrow \{true, false\}$

(representing the capability to deal), and  $right-to-deal : Product, Conditions, Counterpart \rightarrow \{true, false\}$  (representing the right to deal), where  $Product$ ,  $Conditions$  and  $Counterpart$  stand, respectively, for the product to be traded, the conditions associated to that operation and, the counter-part agent involved in the deal. It may now be stated that:

$$\models \forall_{Product} \forall_{Conditions} \forall_{Counterpart} capability - to - deal(Product, Conditions, Counterpart).$$

Therefore, the knowledge about the *capability to deal* is replaced in the *KB* by the *right to deal* ( $right - to - deal : Product, Conditions, Counterpart \rightarrow \{true, false\}$ ).

### 3 Preliminaries

Before approaching each of the system's entities in order to formalize it using *ELP*, the introduction of a series of definitions is in order, namely the ones for *theorem solver*, *restriction/invariant* and *incomplete information*, that pose the basic building blocks for further developments.

A *theorem solver* stands as the mechanism that, making use of the set of logical clauses and inference rules present at the agent's *KB*, is able to determine the logic valuation of a given theorem. A simple theorem solver is now presented for clauses expressed in the traditional *Logic Programming (LP)* language.

#### Definition 1 (LP Theorem Solver)

*Taking factual clauses (represented by  $ag_x : P$ , being read as  $P$  is true for  $ag_x$ ) and rule clauses (represented by  $ag_x : P \leftarrow Q$ , being read as  $P$  if  $Q$  for agent  $ag_x$ ) as the components of the *KB* of agent  $ag_x$ , the predicate  $ag_x : demo_{LP} : T \rightarrow \{true, false\}$ , where  $T$  denotes an abstraction of possible attributes to be found at the predicate's extension, being true and false logical constants that stand for themselves; i.e., the *LP theorem solver* over the *KB* of agent  $ag_x$ , is to be defined in terms of the following set of rules:*

$$\begin{aligned} ag_x : demo_{LP}(true). \\ ag_x : demo_{LP}(ag_x : P \leftarrow Q) \leftarrow ag_x : P \leftarrow Q, ag_x : demo_{LP}(ag_x : Q). \\ ag_x : demo_{LP}(ag_x : P) \leftarrow ag_x : P. \end{aligned}$$

It is now possible to state that factual clauses represented by  $ag_x : P$ , and rule clauses represented by  $ag_x : P \leftarrow Q$ , present in the *KB* of agent  $ag_x$ , may be denoted as  $P$  and  $P \leftarrow Q$ , without loss of meaning, whenever the owner agent is implicit.

A restriction or invariant is a condition or a set of conditions that are to be maintained by an agent on a permanent basis. In the general case, the theory of invariants is captured in terms of the predicates' clauses given by the theory of rules, by way of a meta theoretic definition.

**Definition 2 (Invariants)**

*Invariants are represented at the agent's KB through clauses of the form  $A : +restriction :: P$ , where  $A$ ,  $+restriction$  and  $P$  stand, respectively, for a given agent, the invariant's type and the invariant itself.*

Typically, an agents act in situations where a deal is out of bounds or, in some way, the deal procedures are not completely defined; such situations to be handled properly, involve the use of *null* values. A special theorem solver had, therefore, to be object of development and analysis, in order to cope with this kind of information.

**Definition 3 (LP Theorem Solver for Incomplete Information)**

*Taking factual clauses (represented by  $P$ ) and rule clauses (represented by  $P \leftarrow Q$ ) as the components of the KB of agent  $ag_x$ , the predicate  $demo_{incomplete-information} : T, V \rightarrow \{true, false\}$ , where  $T$  and  $V$  stand, respectively, for an abstraction of the possible attributes to be found at the predicates' extension, and the possible valuations for  $T$ , the LP theorem solver for incomplete information with null values over the KB of an agent, will be given by the following set of rules:*

$$\begin{aligned} demo_{incomplete-information}(T, true) &\leftarrow T. \\ demo_{incomplete-information}(T, false) &\leftarrow \neg T. \\ demo_{incomplete-information}(T, unknown) &\leftarrow not T, not \neg T. \end{aligned}$$

With the use of incomplete information through null values, a simple three-valued logic is set into place. Using this framework, it is now possible to assert the conditions under which a given product or service may be traded.

The situation where the ability to trade product  $P$  with agent  $Y$  is known, but the trading conditions are not defined, may be overcome with the use of a *null value from an unknown set of values* [2].

**Definition 4 (Unknown Conditions taken from an Unknown Set of Values)**

*An unknow condition situation arises when an agent can not express properly in its KB its right to deal with a given counter-part, once the trade conditions are taken from an unknown set of possible values. Taking predicates  $null_{unknown-set-of-values} : N \rightarrow \{true, false\}$  (expressing the null value statement),  $right-to-deal : P, C, CP \rightarrow \{true, false\}$  (expressing the agent's right to deal) and  $exception_{right-to-deal} : P, C, CP \rightarrow \{true, false\}$  (expressing an exception on the agent's right to deal), where  $N$ ,  $P$ ,  $C$  and  $CP$  stand, respectively, for a null value, the product to be traded, the conditions to consider during negotiation, and the counterpart agent, then the KB of an agent is to be augmented by the rules:*

$$\begin{aligned} exception_{right-to-deal}(P, -, CP) &\leftarrow null_{unknown-set-of-values}(X), \\ &\quad right-to-deal(P, X, CP). \\ \neg right-to-deal(P, C, CP) &\leftarrow not right-to-deal(P, C, CP), \\ &\quad not exception_{right-to-deal}(P, C, CP). \end{aligned}$$

*where predicate  $\neg right-to-deal : P, C, CP \rightarrow \{true, false\}$  stands for an abstraction of the predicate's extension on the situations under which  $P$  is untradable, under the market conditions  $C$  and counterpart agent  $CP$ . The KB of an agent must contain an extension of predicates  $null_{unknown-set-of-values}()$  (e.g.,  $null_{unknown-set-of-values}(somecondition)$ )*

and of  $right - to - deal()$  (e.g.,  $right - to - deal(p4, somecondition, cp2)$ ).

Extending the previous definition of incompleteness, it can be observed that a new trade situation, in terms of a different set of market conditions, is possible. This new situation arises when one intends to trade product  $P$  with agent  $Y$ , although the set of conditions under which the operation is to be carried out is unknown; i.e., are taken from a pre-defined set of values. This scenario can be set by using disjunctive information [2].

**Definition 5 (Unknown Conditions taken from a Well-Known Set of Values)**

*This type of unknown condition situation arises when an agent can not express conveniently in its KB its right to deal with a given counter-part, once the trade conditions, although unknown are taken from a well-known set of possible values. Taking predicates  $null_{known-set-of-values} : N \rightarrow \{true, false\}$ ,  $conditions : P, CP, C \rightarrow \{true, false\}$  (expressing a valid set of trade conditions),  $right - to - deal : P, C, CP \rightarrow \{true, false\}$  (expressing the agent's right to deal) and  $exception_{right-to-deal} : P, C, CP \rightarrow \{true, false\}$  (expressing an exception on the agent's right to deal), where  $N, P, C$  and  $CP$  stand, respectively, for a null value, the product to be tradable, the conditions to consider during negotiation, and the counterpart agent, then the KB of an agent is augmented by the rules:*

$$\begin{aligned}
 exception_{right-to-deal}(P, C, CP) &\leftarrow null_{known-set-of-values}(X), \\
 &\quad conditions(P, CP, C), \\
 &\quad right - to - deal(P, X, CP). \\
 \neg right - to - deal(P, C, CP) &\leftarrow not\ right - to - deal(P, C, CP), \\
 &\quad not\ exception_{right-to-deal}(P, C, CP).
 \end{aligned}$$

*where predicate  $\neg right - to - deal : P, C, CP \rightarrow \{true, false\}$  stands for an abstraction of the predicate's extension on the situations under which  $P$  is untradable, under the market conditions  $C$  and counterpart's agent  $CP$ . The KB of an agent must contain an extension of predicates  $null_{known-set-of-values}()$  (e.g.,  $null_{known-set-of-values}(some-defined-condition)$ ),  $conditions()$  (e.g.,  $conditions(p4, cp2, [c1, c2])$ ) and  $right - to - deal()$  (e.g.,  $right - to - deal(p4, some - defined - condition, cp2)$ ).*

Querying the KB of an agent on its  $right - to - deal$  takes, therefore, the form of the following instance:  $\neg demo_{incomplete-information}(right - to - deal(p4, [c1, c2], cp3))$

To represent the fact that an agent is inhibited to trade product  $P$  with an agent  $Y$ , the use of a *null value* of the type *not permitted* is appropriate [2]. This kind of situation is well expressed by combining  $right - to - deal$  clauses with the *null* statement and exception clauses.

**Definition 6 (Non Permitted Trade Situation)**

*A non permitted trade situation arises when an agent can not express properly in its KB its right to trade with some other counter-part, once such privilege was not allocated to it. Taking predicates  $null_{not-permitted} : N \rightarrow \{true, false\}$  (expressing the null value statement),  $right - to - deal : P, C, CP \rightarrow \{true, false\}$  (expressing the agent's right to deal) and  $exception_{right-to-deal} : P, C, CP \rightarrow \{true, false\}$  (expressing an exception on the agent's right to trade), where  $N, P, C$  and  $CP$  stand, respectively, for a null value, the product to be tradable, the trade's conditions to pursue during negotiation and the counterpart agent, then the agent's KB of an agent is augmented by the rules:*

$$exception_{right-to-deal}(P, C, -) \leftarrow \begin{array}{l} null_{not-permitted}(X), \\ right-to-deal(P, C, X). \end{array}$$

$$assert_{right-to-deal}(P, C, CP) \leftarrow \begin{array}{l} not\ right-to-deal(P, C, CP), \\ not\ exception_{right-to-deal}(P, C, CP), \\ assert(right-to-deal(P, C, CP)). \end{array}$$

where predicate  $assert_{right-to-deal} : P, C, CP \rightarrow \{true, false\}$  stands for the controlled assertion of the clause  $right-to-deal(P)$ , under tradable's conditions  $C$ , being  $CP$  the counterpart's agent. The  $KB$  of an agent must contain an instantiation of  $null_{not-permitted}$  (e.g.,  $null_{not-permitted}(someone)$ ) and  $right-to-deal()$  clauses which may use a null value (e.g.,  $right-to-deal(pA, [c1], someone)$ ).

## 4 The Logic Structure Behind Negotiation

As it was previously stated, the set of the most important features that intervene in the negotiation process, will now be object of study. These include features such as temporality, priority, delegation, gratitude and agreement. Through assessment, an agent may evaluate its  $KB$ , its temporal validity and relative priorities, in order to decide if delegation is to be pursued, if gratitude debts are established or any kind of socialization is to take place.

The general process of *negotiation* must be clearly distinguished from the *argumentative* one [4]. The process of argumentation is tightly coupled with the process of logically founded attack on the arguments put forward by a counterpart. It works with price-formation and trade issues. On the other hand, negotiation is a wider concept that is coupled with specific forms of reasoning, dealing with the high-order, pre-argument relationships, that may be established among agents.

In order to establish a common ground for the formalization of each concept, a logical theory (on which the  $KB$  of each agent is based upon) is to be defined, which will serve as the backbone of the negotiation process.

### Definition 7 (Logical Theory for Negotiation Agents)

A logical theory for negotiation agents is defined as the quadruplet  $TNA = \langle R, C, BP, \prec \rangle$  where  $R$ ,  $C$ ,  $BP$  and  $\prec$  stand, respectively, for an abstraction of the extension of the predicate's  $right-to-deal$  ( $right-to-deal : Product, Conditions, Counterpart \rightarrow \{true, false\}$ ), the set of invariants ( $A : +restriction :: P$ ), the set of behavioural predicates (including the theorem proffers) and a non-circular order relation that states that if  $P \prec Q$ , then  $P$  occurs prior to  $Q$ .

### 4.1 Temporality

The concept of temporality is linked to the agent's states of knowledge. Taking a non-destructive  $KB$ , different conclusions may be attained at different points in time (e.g., John may deal with

Paul but only from 10/05/2000 to 12/05/2000).

The set  $R$ , understood as it was stated above, as an abstraction of the extension of the predicate *right-to-deal* has to be reconstruct in order to fulfill temporal validity (Figure 1).

**Definition 8 (Clauses with Temporality)**

A factual clause, given as  $P$  is represented, to encompass temporal validity, as  $P :: [i_1, i_2, \dots, i_n]$ , where  $i_j = [t_a, t_b]$ , stands for:

- *temporal instant*  
 $t_a = t_b, t_a, t_b \geq 0$  with  $t_a, t_b \in TD$ .  
 $TD = \{t | t \in N_0\} \cup \{\text{forever}\}$ , where *forever* represents the end of times.
- *temporal interval*  
 $t_a < t_b, t_a \geq 0$  with  $t_a, t_b \in TD$ .  
 $TD = \{t | t \in N_0\} \cup \{\text{forever}\}$ , where *forever* represents the end of times.

where  $\exists P :: T_1 \wedge \exists P :: T_2 \rightarrow T_1 \cap T_2 = \emptyset$

Having restructured the data present in the agent's *KB*, a theorem solver has to be constructed in order to deal with the new situation. This theorem solver is taken as an evolution of the one used to deal with *incomplete information*, referred to above.

**Definition 9 (LP Theorem Solver for Incomplete and Temporal Information)**

Taking factual clauses in the form  $P :: [i_1, i_2, \dots, i_n]$  and rule clauses (represented in the form  $P \leftarrow Q$  and being read as *P if Q*) as the components of the agent's *KB*, the predicate  $\text{demo}_{LPIT} : T, CT, V \rightarrow \{\text{true}, \text{false}\}$ , where  $T, CT$ , and  $V$  stand, respectively, for a logical theorem, the current time, a logical constant taken from the set  $\{\text{true}, \text{false}, \text{unknown}\}$ , the LP theorem prover for incomplete and temporal information over the *KB*, is therefore given by the following set of rules:

$$\begin{aligned} \text{demo}_{LPIT}(P, CT, \text{true}) &\leftarrow P :: T, \text{in}_{time}(CT, T). \\ \text{demo}_{LPIT}(P, CT, \text{false}) &\leftarrow P :: T, \neg \text{in}_{time}(CT, T). \\ \text{demo}_{LPIT}(P, CT, \text{false}) &\leftarrow \neg(P :: T), \text{in}_{time}(CT, T). \\ \text{demo}_{LPIT}(P, -, \text{unknown}) &\leftarrow \text{not}(P :: -), \text{not}(\neg(P :: -)). \end{aligned}$$

$$\begin{aligned} \text{in}_{time}(CT, [T]) &\leftarrow \text{in}_{interval}(CT, T). \\ \text{in}_{time}(CT, [T_1|T]) &\leftarrow \text{in}_{interval}(CT, T_1), \text{in}_{time}(CT, T). \end{aligned}$$

$$\begin{aligned} \text{in}_{interval}(CT, [T_a, T_b]) &\leftarrow T_a \leq T_b, T_a \geq 0, CT \geq T_a, CT \leq T_b. \\ \text{in}_{interval}(CT, [T_a, \text{forever}]) &\leftarrow T_a \geq 0, CT \geq T_a. \end{aligned}$$

where predicates  $\text{in}_{time} : CT, LT \rightarrow \{\text{true}, \text{false}\}$  and  $\text{in}_{interval} : CT, TI \rightarrow \{\text{true}, \text{false}\}$  state that  $CT$  stands both for values in the time interval  $LT$  and the values  $TI$ , respectively.

## 4.2 Priorities

As it was previously stated, the logical theory *TNA* introduces a non-circular ordering through relation  $\prec$  on the clauses that make the agent's *KB*. However, this ordering, based on a temporal sequence of the *KB* assertions, is insufficient to express, in particular, some *right-to-deal* operations; indeed, real-world agents (e.g., humans beings) are able to prioritize the treatment



of knowledge in their reasoning processes. As an example, consider the following set of factual clauses that make the *KB* of an agent:

$$\begin{aligned} &right\_to\_deal(p1, somecondition, john). \\ &right\_to\_deal(p1, [c1], charles). \\ &right\_to\_deal(p1, [c1, c2], paul). \end{aligned}$$

Having in mind that the *KB* of an agent is supposed to be an ordered, non-circular, logical theory *TNA*, this set of facts can be interpreted, in terms of the relative ordering of clauses, as: there is the right to deal product *p1*, under unspecified pre-conditions, with *John*, then it will be possible to deal product *p1*, under pre-condition *c1*, with *Charles* and, the last but not the least, it is possible to deal product *p1*, under pre-conditions *c1* and *c2*, with *Paul*. Any change on the priority treatment of clauses (e.g., having the clauses that relate to *Paul* with higher priority than those that relate to *John*), would lead to a restructuring of the *KB*, according to the relation  $\prec$ , in order to express the new priority semantics. However, this course of action is unfeasible in systems where large sets of clauses are present.

A reliable approach, for a feasible priority treatment of the agent's clauses, relies on the embedding of priority rules on the *KB* of each agent. Therefore, a logical theory *TNA* is to be changed into a new one, *TNAP*, in which the grouping of the factual clauses is performed according to the semantics of the priority rules.

**Definition 10 (Logical Theory for Negotiation Agents with Priorities)**

*A logical theory for negotiation agents with priorities is defined as  $\langle R, C, BP, PR, \prec \rangle$  where,  $R, C, BP, PR$  and  $\prec$  stand, respectively, for the set of predicates on the right to deal (e.g.,  $right\_to\_deal:Product, Conditions, Counterpart \rightarrow \{true, false\}$ ), the set of asserted restrictions or invariants ( $A : +restriction :: P$ ), the set of behavioral predicates, the set of embedded priority rules and a non-circular order relation established among the different clauses in the *KB*. Relation  $\prec$  sets, in the case  $P \prec Q$ , that  $P$  is earlier than  $Q$ , thus ordering on the set of clauses, providing for a fail-safe priority mechanism.*

The use of precedences in the human reasoning process has been a question of common sense. Although priorities can be established between single clauses, it is usual to consider priorities among *knowledge bodies* (e.g., stating that information about *Mary* has priority over information about *John*). These *knowledge bodies* denote nothing more than an high-level classification of the factual clauses. Notice, however, that this classification has variable granularity, giving way to the assertion a per-clause priority if needed (with the consequent increase in computational complexity). Therefore, the factual clauses dealing with the agent's right to deal, already expanded to include temporal information, are now set to include a *body of knowledge* classification. Reasoning can now take place without *hard-ordering* the clauses.

**Definition 11 (Clauses with Temporality and a Body of Knowledge Classification)**

Taking  $P :: [i_1, i_2, \dots, i_n]$  as a clause, where  $P$  is the simple factual clause and  $[i_1, i_2, \dots, i_n]$  stands for its temporal validness, then  $BK :: P :: [i_1, i_2, \dots, i_n]$  denotes a clause with temporality and a body of knowledge classification, where  $BK$  stands for the a body of knowledge to which  $P$  is associated.

Having defined the concept of *body of knowledge* and the structure of the factual clauses that express the right to deal, the structure of the  $PR$  rules in theory  $TNAP$  may now be stated.

**Definition 12 (Priority Clauses)**

Taking  $BK_i$  and  $BK_j$  ( $i \neq j$ ) denote two bodies of knowledge, and that the knowledge associated with  $BK_i$  has priority over the knowledge associated with  $BK_j$ ; then the procedure to be insert into the set  $PR$  of the logical theory  $TNAP$ , is given by  $priority : BK_i, BK_j \rightarrow \{true, false\}$ . Set  $PR$  should be consistent; i.e., it has to fulfil the invariant:

$$\begin{aligned} (\forall BK_i, BK_j), priority(BK_i, BK_j) &\rightarrow \neg successor(BK_j, BK_i) \\ successor(X, Y) &\leftarrow priority(X, Y). \\ successor(X, Y) &\leftarrow priority(X, Z), successor(Z, Y). \end{aligned}$$

In order to use the priority information in the reasoning processes at the agent's level, the theorem prover presented above to deal with incomplete and temporal information has to be rewritten.

**Definition 13 (LP Theorem Prover for Incomplete and Temporal Information with Priorities)**

Taking factual clauses with temporal validity, a body of knowledge classification (represented by  $BK :: P :: [i_1, i_2, \dots, i_n]$ ) and rule clauses (represented by  $P \leftarrow Q$ ) as the components of the  $KB$  present in each agent, the predicate  $demo_{LPITP} : T, CT, V \rightarrow \{true, false\}$ , where  $T, CT, V$  and  $\{true, false\}$  stands, respectively, for a logical theorem, the current time, the theorem valuation ( $true, false$ ), the  $demo_{LPITP}$  predicate denotes the theorem solver for incomplete and temporal information over  $KB$ , in terms of the following set of rules:

$$\begin{aligned} demo_{LPITP}(P, CT, true) &\leftarrow priority(BK_1, BK_2), \\ &\quad test_{priority}(BK_1, BK_2, P, T), \\ &\quad in_{time}(CT, T). \\ demo_{LPITP}(P, CT, false) &\leftarrow priority(BK_1, BK_2), \\ &\quad test_{priority}(BK_1, BK_2, P, T), \\ &\quad \neg in_{time}(CT, T). \\ demo_{LPITP}(P, CT, false) &\leftarrow priority(BK_1, BK_2), \\ &\quad ntest_{priority}(BK_1, BK_2, P, T), \\ &\quad in_{time}(CT, T). \\ demo_{LPITP}(P, -, unknown) &\leftarrow priority(BK_1, BK_2), \\ &\quad not test_{priority}(BK_1, BK_2, P, -), \\ &\quad not ntest_{priority}(BK_1, BK_2, P, -). \\ \\ test_{priority}(BK_1, -, P, T) &\leftarrow BK_1 :: P :: T. \\ test_{priority}(-, BK_2, P, T) &\leftarrow BK_2 :: P :: T. \\ \\ ntest_{priority}(BK_1, -, P, T) &\leftarrow \neg(BK_1 :: P :: T). \\ ntest_{priority}(-, BK_2, P, T) &\leftarrow \neg(BK_2 :: P :: T). \end{aligned}$$

where predicates  $in_{time} : CT, LT \rightarrow \{true, false\}$ ,  $test_{priority} : BK_a, BK_b, P, T \rightarrow \{true, false\}$  and  $ntest_{priority} : BK_a, BK_b, P, T \rightarrow \{true, false\}$  stand for themselves.

### 4.3 Delegation

Delegation is to be understood as the endorsement of negotiation tasks to an agent or group chosen to represent another or others. Negotiation tasks may only be assigned to a third party if there is sufficient information to assess the right to deal of such peers.

Delegation, as hereby presented, opens the way to undertake indirect negotiations; i.e., to use proxy agents, taking advantage of their soul, such as gratitude debts or agreements that may have been established among the proxy with another or other agents. The delegation process is equivalent to the generation of a *middle-man* approach to business.

#### Definition 14 (Delegation)

Taking logical theory *TNAP* over which the KB of an agent is built and the theorem solver for incomplete information with priorities  $demo_{LPITP} : P, CT, V \rightarrow \{true, false\}$ , agent  $ag_x$  can delegate the negotiation on agent  $Ag$ , in order to have it acting as a proxy element, as long as its reasoning is regulated by the logical structure of predicate *delegate* : *What, Conditions, CounterPart, Ag, CurrentTime*  $\rightarrow \{true, false\}$ . This situation is materialized by the set of rules:

$$\begin{aligned}
 ag_x : delegate(P, C, CP, Y, CT) \leftarrow \\
 & ag_x : demo_{LPITP}(right - to - deal(P, C, CP), CT, true), \\
 & ag_x : demo_{LPITP}(right - to - deal(P, -, Y), CT, true), \\
 & Y : valid_{assimilation}(Y : right - to - deal(P, C, CP), CT).
 \end{aligned}$$

Then,  $\forall ag_y$  such that  $ag_y \in Agents$ , it follows that:

$$\begin{aligned}
 ag_y : valid_{assimilation}(X, CT) \leftarrow ag_y : demo_{LPITP}(X, CT, true). \\
 ag_y : valid_{assimilation}(X, CT) \leftarrow & ag_y : assert(X :: [[CT, CT]]), \\
 & ag_y : solutions(Z, ag_y : +restriction :: Z, S), \\
 & ag_y : demo_{restrictions}(S, CT, true). \\
 ag_y : valid_{assimilation}(X, CT) \leftarrow & ag_y : retract(- :: X :: [[CT, CT]]). \\
 ag_y : demo_{restrictions}([], -, true). \\
 ag_y : demo_{restrictions}([R_1], CT, V) \leftarrow & ag_y : demo_{LPITP}(R_1, CT, V). \\
 ag_y : demo_{restrictions}([R_1|T], CT, true) \leftarrow & ag_y : demo_{LPITP}(R_1, CT, true), \\
 & ag_y : demo_{restrictions}(T, CT, true). \\
 ag_y : demo_{restrictions}([R_1|T], CT, false) \leftarrow & ag_y : demo_{LPITP}(R_1, CT, V_1), \\
 & ag_y : demo_{restrictions}(T, CT, V_2), V_1 \neq V_2.
 \end{aligned}$$

where predicates  $valid_{assimilation} : X, CT \rightarrow \{true, false\}$ ,  $solutions : A, C, S \rightarrow \{true, false\}$ , and  $demo_{restrictions} : S, CT, V \rightarrow \{true, false\}$  stand for themselves.

### 4.4 Gratitude

In real-life, the establishment of relationships among human beings, especially those of a commercial nature, have to deal with non-linearities such as *gratitude*. Gratitude is seen as the credit put into the establishment of a relationship among agents (e.g., with agents  $ag_x$  and  $ag_y$ , deriving from a process of explicit or implicit debts, where there is an obligation to pay or perform something).

Although gratitude quantification is possible and even desirable in order to cater for computational manipulation, it is still a subjective element, an element that has decisive influence in the outcome of many business strategies (e.g., in strategic planning). Gratitude may derive from one of two main situations: the use of offerings that introduces non-linearities in the negotiation process or a situation of drop-out (i.e., end-of-competition).

On the other hand, one has to consider the concept of *marginal gratitude* in order to formalize *gratitude*. *Marginal gratitude* is a quantified representation of the gratitude debt of an agent  $ag_x$  towards another agent  $ag_y$ , taking into account the particular aspects of a given negotiation. In [3] it is possible to find a formal characterization of a quantification of the *marginal gratitude* concept, in terms of the function  $gm()$ , depicted below:

$$g_m(x, y, NI) = \begin{cases} value_{offer} - W_{strategy}(NI, y), & NI.grt = non - negotiable \\ (1 - \alpha)F_{gains}(NI), & NI.grt = negotiable \\ 0, & otherwise \end{cases}$$

where  $g_m(x, y, NI)$ ,  $W_{strategy}(NI, y)$ ,  $value_{offer}$ ,  $\alpha$  and  $F_{gains}(NI)$  stand, respectively, for the marginal gratitude of agent  $x$  towards another agent  $y$ , taking into consideration the market information  $NI$ ; the function that weights the influence of  $NI$  and agent  $y$  in the strategy of another agent  $x$ ; the credit attributed by agent  $x$  to the offer that developed a gratitude debt; the percentage of the gains to offer to agent  $y$  as a compensation for a drop-out; and the forecast of the gains taking into account  $NI$ .  $NI$  is a composite data structure that includes fields such as the kind of gratitude (*grt*).

### Definition 15 (Gratitude)

Considering the solutions space given by predicate  $demo_{LPITP} : P, CT, V \rightarrow \{true, false\}$ , it is possible to state that agent  $ag_x$  has a debt of gratitude towards agent  $Ag$ , due to a specific offer or due to a drop-out situation, as long as its reasoning proceedings are regulated by the logical structure of predicate  $gratitude : Value, NI, Ag, CurrentTime \rightarrow \{true, false\}$ , given in terms of the productions depicted below:

$$\begin{aligned} ag_x : gratitude(Value, NI, Y, CT) \leftarrow & \\ ag_x : offer(Y, Description), & \\ ag_x : demo_{LPITP}(right - to - deal(-, -, Y), CT, true), & \\ ag_x : evaluate_{offer}(Y, Description, CT, Value_{offer}), & \\ ag_x : update(NI, Y, Description, Value_{offer}, CT, NNI), & \\ ag_x : evaluate_{strategy}(NNI, Y, CT, S, S_{weight}), & \\ ag_x : gratitude_{marginal}(NNI, [Value_{offer}, S_{weight}], Value), Value > 0. & \end{aligned}$$

$$\begin{aligned} ag_x : gratitude(Value, NI, Y, CT) \leftarrow & \\ ag_x : demo_{LPITP}(right - to - deal(-, -, Y), CT, true), & \\ Y : drop_{compensation}(ag_x, NI, CT, \alpha), & \\ ag_x : forecast_{gains}(Y, NI, CT, F), & \\ ag_x : gratitude_{marginal}(NI, [\alpha, F], Value), Value > 0. & \end{aligned}$$

where predicates  $evaluate_{offer} : Y, D, CT, V \rightarrow \{true, false\}$ ,  $update : NI, Y, D, V, CT, NNI \rightarrow \{true, false\}$ ,  $evaluate_{strategy} : NNI, Y, CT, S, S_{weight} \rightarrow \{true, false\}$ ,  $gratitude_{marginal} : NI, P, V \rightarrow \{true, false\}$ ,  $drop_{compensation} : X, NI, CT, \alpha \rightarrow \{true, false\}$  and  $forecast_{gains} : Y, NI, CT, F \rightarrow \{true, false\}$  state, respectively, that the valuation of offer value  $V$ , on the part of agent  $ag_x$  to any offer made by agent  $Y$ , with description  $D$ , at time  $CT$ ; the updating of negotiation information  $NI$  ( $NNI$ ), taking counter-part agent  $Y$ , the offer description  $D$ , the offer value  $V$ , at time  $CT$ ; the  $S$  strategy valuation ( $S_{weight}$ ) of agent  $ag_x$  with negotiation information  $NI$ , counter-part agent  $Y$ , at time  $CT$ ; the marginal gratitude ( $V$ ) taking into account negotiation information  $NI$  and the set of parameters  $P$ ; the compensation ( $\alpha$ ) for the drop-out of the counter-part of agent  $X$ , in a deal with negotiation information  $NI$ , at time  $CT$ ; and the gains forecast ( $F$ ), taking counter-part agent  $Y$ , negotiation information  $NI$ , at time  $CT$ .

## 4.5 Agreement

Inside an organization, one may see the need for agreement, but when using *collaborative* agents, one is faced with a truthful environment. In these environments one is able to formalize agreements in order to provide an integrated picture of the system to the outside world. The agents are assumed to be fully veridic – telling or expressing the truth – they make their own way on a basis of beliefs in their own abilities; *i.e.*, their qualities to cope with a problem. This image can be achieved by gathering all the agent’s opinions on a particular subject and the use of a majority vote. The agreement strategy can, therefore, be rewritten in order to weight each agent’s specificity. This may be expressed as:

$$agreement_w(value) = majority[ w_1 opinion_1(value), \\ w_2 opinion_2(value), \\ \dots, \\ w_n opinion_n(value) ] \\ , value \in \wp \quad \text{with } 0 \leq w_i \leq 1, \forall i \in Agents$$

where  $w_i$  and *Agents* stand, respectively, for the weight of agent  $i$  in the making of *value*, and the community of agents  $w_i$ , on the other hand, may be a function of time, which can be expressed as:  $w_i(t) = \beta w_i(t-1) + (1-\beta)compliance(v_{p,i}(t), v_{r,i}(t))$  with  $0 \leq \beta \leq 1$  where  $\beta$ ,  $v_{p,i}(t)$ ,  $v_{r,i}(t)$  and  $compliance(x, y)$  stand, respectively, for the weight of the historic information in the making of  $w_i(t)$ ; the judgment of agent  $i$  at time  $t$  on the *value*; the *value* attained by agent  $i$  at time  $t$ ; and a measure of the reciprocity or relation between  $v_{p,i}(t)$  and  $v_{r,i}(t)$ . The higher the value of  $\beta$  the higher the significance of historic values and smaller the influence of sporadic non-compliances. The use of non-veridic agents introduces the need for special opinion-selection criteria.

In logical terms, an agreement can only be reached among agents that are able to deal with each other; *i.e.*, if an agent is unable to assert the right to deal with another or other agents, it

can never establish some sort of commitment (agreement).

**Definition 16 (Agreement)**

Considering the logical theory *TNAP* over which the *KB* of an agent is built and the theorem prover for incomplete and temporal information with priorities  $demo_{LPITP} : P, CT, V \rightarrow \{true, false\}$ , it may be stated that agent  $ag_x$  has a debt of gratitude towards agent  $Ag$ , due to a specific offer or due to a drop-out situation, as long as its reasoning proceedings are regulated by the logical structure of predicate agreement : *Subject, Elements, Opinion, CurrentTime*  $\rightarrow \{true, false\}$ , whose extension is given in terms of the productions:

$$\begin{aligned}
 ag_x : agreement(-, [ag_x], -, -). \\
 ag_x : agreement(S, E, O, CT) \leftarrow & \quad ag_x : can - deal - with(E, CT), \\
 & \quad ag_x : gather_{opinions}(S, E, CT, LO), \\
 & \quad ag_x : summarize(S, O, LO, CT). \\
 \\
 ag_x : can - deal - with([A], CT) \leftarrow & \\
 & \quad ag_x : demo_{LPITP}(right - do - deal(-, -, A), CT, true). \\
 ag_x : can - deal - with([A|T], CT) \leftarrow & \\
 & \quad ag_x : demo_{LPITP}(right - do - deal(-, -, A), CT, true), \\
 & \quad ag_x : can - deal - with(T, CT).
 \end{aligned}$$

where predicates *can-deal-with* :  $L, CT \rightarrow \{true, false\}$ , *gather\_opinions* :  $S, L, CT, LO \rightarrow \{true, false\}$  and *summarize* :  $S, O, LO, CT \rightarrow \{true, false\}$ , state, respectively, the legal authority of agent  $ag_x$  to deal with the set of agents  $L$ , at time  $CT$ ; the process of opinions ( $LO$ ) gathering on the agreement on subject  $S$  by the agents that make the set  $L$ , at time  $CT$ ; and the final word on the agreement on subject  $S$ , taking the opinion of agent  $ag_x$  ( $O$ ), the list of the other agent's opinions ( $LO$ ), and the current time  $CT$ .

## 5 Some Examples

Based on the bulk of knowledge defined to above, the consideration of a small set of examples, dealing with the different *right-to-deal* situations in which an agent is involved, is paramount. Therefore, in order to continue with this study, it is used the agents *KB* defined according to the non-circular logic theory *TNAP* depicted below, in terms of the productions:

$$\begin{aligned}
 & null_{forbidden}(someone). \\
 & null_{unknown-set}(somecondition). \\
 bk_1 :: right - to - deal(p1, somecondition, cp1) :: & \quad [[0, 5], [8, forever]]. \\
 bk_1 :: \neg right - to - deal(p1, somecondition, cp1) :: & \quad [[6, 7]]. \\
 bk_2 :: right - to - deal(p1, [c1], cp2) :: & \quad [[0, forever]]. \\
 bk_2 :: right - to - deal(p3, [c1, c2], cp2) :: & \quad [[0, 10], [30, 40]]. \\
 bk_3 :: right - to - deal(p5, [c1], someone) :: & \quad [[0, forever]]. \\
 bk_3 :: right - to - deal(p6, [c1, c3], someone) :: & \quad [[0, forever]]. \\
 \%exceptions \\
 \%theorem\ provers \\
 priority(bk_2, bk_1). \\
 priority(bk_1, bk_3).
 \end{aligned}$$

The agent's *KB* endorses two types of null values: a null value of the type *not permitted* and

a null value of the type *value unknown and not necessarily from a given set of values*. Therefore, products  $p5$  and  $p6$ , taking into account the semantics of the null of the type *not permitted* (given the conditions  $c1$ , and  $c1$  and  $c3$ ), have their dealings conditioned (i.e., in this case, an agent can not trade these products with any peer). On the other hand, product  $p1$  may be traded with counter-part agent  $cp1$ , however, due to a null of the type *value unknown and not necessarily from a given set of values*, the set of invariants that delimit the borders of the negotiation is not stated (e.g., it can not be said that this product can be traded under conditions  $c1$  and  $c3$ , once this fact is unknown).

The negotiation of product  $p1$  with counter-part agent  $cp1$  develops across three time intervals. Two of these time intervals incorporate positive information (i.e., the right to deal exists), but one of them is fitted with negative information (i.e., the right to deal does not exist). In the case of product  $p3$ , to be negotiated with counter-part agent  $cp2$  under trade conditions  $c1$  and  $c2$ , two time intervals were defined where the right to deal exists. Outside these intervals, such a right is unknown.

Three *bodies of knowledge* have been defined taking into account the counter-part agent; i.e., although the productions that make the extension of the predicate *right-to-deal* with respect to agent  $cp1$  appear in the agent  $KB$  prior to those of agent  $cp2$ , and those agent's of  $cp2$  prior to those of an unspecified agent (i.e., agent's *someone*), the clauses' priority must be object of consideration (i.e., in this case, in any inference process, the theorem prover will invoke in first place those clauses linked to agent  $cp2$ , then those to an unspecified agent (here denoted as *someone*), and finally those linked to agent  $cp1$ ). The agent's  $KB$  may now be queried through the valuation of the logic formulae (i.e., theorems), in the form:

$$\begin{aligned}
(\forall (V)), \neg dem_{LPITP}(right - to - deal(p1, [c1], cp2), 10, V) & \Leftrightarrow V = true \\
(\forall (V)), \neg dem_{LPITP}(right - to - deal(p5, [c1], cp3), 10, V) & \Leftrightarrow V = unknown \\
(\forall (V)), \neg dem_{LPITP}(right - to - deal(p1, somecondition, cp1), 7, V) & \Leftrightarrow V = false \\
(\forall (P, C, CP)), \neg dem_{LPITP}(right - to - deal(P, C, CP), 10, true) & \\
\Leftrightarrow P = [p1, p3, p1, p5, p6] & \\
C = [[c1], [c1, c2], somecondition, [c1], [c1, c3]] & \\
CP = [cp2, cp2, cp1, someone, someone] &
\end{aligned}$$

In the first and third queries, the formulae's valuation is reached, respectively, through the use of positive and negative information presented in the KB. The second query is valued to *unknown* due to the use of null values in the set of conditions; i.e., product  $p5$  can be negotiated with agent  $cp3$  but the set of conditions can not be set to  $c1$ , although it is a possible value. The last query indicates the possible variable instantiations of product, conditions and counter-parts, taking into

account that the right to deal must be stated as prevailing, in the KB, at instant 10.

Continuing our study, let us consider a new set of clauses representing the KB of agent  $ag_x$ , and that of agent  $ag_y$ . Reasoning about *delegation* will now involve the consideration of the set of restrictions embedded into the KBs, and defined in terms of the productions given below:

```

 $ag_x : bk_1 :: right - to - deal(p1, [c1], cp2) :: [[0, forever]].$ 
 $ag_x : bk_2 :: right - to - deal(p2, [c3, c4], cp3) :: [[0, 50]].$ 
 $ag_x : bk_2 :: right - to - deal(-, -, ag_y) :: [[0, forever]].$ 
%exceptionsagx
%theorem proversagx
 $ag_x : priority(bk_1, bk_2).$ 

 $ag_y : bk_1 :: right - to - deal(p2, [c3, c4], cp3) :: [[0, 60]].$ 
 $ag_y : bk_2 :: right - to - deal(p2, [c5], cp4) :: [[0, 10]].$ 
 $ag_y : money(900).$ 
 $ag_y : +restriction :: money(1000).$ 
%exceptionsagy
%theorem proversagy
 $ag_y : priority(bk_1, bk_2).$ 

```

that state that Agent  $ag_x$  is in the position to negotiate product  $p1$ , under trade conditions  $c1$  with counter-part agent  $ag_y$ , in a permanent base. In the case of product  $p2$ , trade conditions  $c3$  and  $c4$  are established for counter-part agent  $cp3$ , but only on the time interval  $[0,50]$ . The information available about the right to deal with  $ag_y$  overpowers the information available about  $cp3$ .

Agent  $ag_y$  may negotiate product  $p2$ , under trade conditions  $c3$  and  $c4$  with counter-part agent  $cp3$ , but only on the time interval  $[0,60]$ . Furthermore, it may also negotiate product  $p2$ , under trade conditions  $c5$  with counter-part agent  $cp4$ , but only on the time interval  $[0,10]$ . To agent  $ag_y$  were inputted 900 monetary units and, an update of this value is restrained by the invariant  $+restriction :: money(1000)$ , that sets the upper bound of the monetary units domain. Priority rules establish that the information available about  $cp3$  overpowers that of  $cp4$ . The KB may now be queried through the valuation of the logic formulae given below, in order to assess the validity of the delegation processes, in the form:

$$\neg ag_x : delegate(p1, [c1], cp2, ag_y, 10). \Leftrightarrow false$$

$$\neg ag_x : delegate(P, C, cp3, ag_y, 10). \Leftrightarrow P = [p2] \ C = [c3, c4]$$

In the first query, although the right to deal has been well established to agent  $ag_x$ , it is impossible to include into the proxy agent  $ag_y$  the necessary trade's information to allow any deal to go ahead; indeed, the invariant  $ag_y : +restrictions :: money(1000)$  under the present



trade conditions is not satisfied (i.e., valued to the logical constant *true*). In the second query, the delegation to agent *ag<sub>y</sub>* of the right to negotiate with agent *cp3*, at time instant 10, is only possible for product *p2* and under trade conditions *c3* and *c4*.

## 6 Related Work

Interesting developments in the area of legal argumentation through logic have been obtained by researchers as Prakken and Sartor [13], [15], Kowalski and Toni [6], [7]. Work has also been done to extend the principles of legal argumentation to the specific field of *EC* by Brito, Novais and Neves [4]. Indeed, the concept of *rights* is central to the behaviour of an agent. *Rights* in multiagent systems have been approached by the work of Norman, Sierra and Jennings [9].

Negotiations in the area of *Economics* must take into account incomplete information, where it is possible to report some work done by Analide and Neves [1]. This work was extended to the *EC* field by work done by Brito, Novais and Neves [5], [12].

## 7 Conclusions

Logic reveals itself as a simple and powerful tool for process formalization. The use of Logic Programming and, furthermore, the use of Extend Logic Programming provides a mathematical background to otherwise subjective statements. Such tools supply the means for a formal proof of statements over the Knowledge Base of each agent, and the possibility to create working prototypes in feasible times. Although some work has been done in the area of argument formalization, the phase that precedes such process is many times ignored. Before entering an argumentative procedure, an agent must reason over a set of other possible actions. Therefore, this phase must be taken into consideration, correctly formalized, and materialized in terms of a set of processes that are recurrent and deal with issues such that incomplete information, temporality, priorities among statements, just to name a few.

For an agent, mathematical logic can be used to establish a feasible and justifiable negotiation scenario. However, first-order logic does not convey some of the important characteristics of a real-world business, such as the ability to set deals under incomplete information, can be easily approached with the use of null values. Furthermore, an agent must be able to deal with knowledge that is not continuous (i.e., the knowledge is only valid in specific time periods) and that is prioritized (i.e., when questioning the Knowledge Base of an agent, variable instantiations are

given taking into account the order of the clause in the agent knowledge base – the priority)). The basic elements of the reasoning procedure must be established (the *right to deal*) and the inference mechanisms over them defined. These inference mechanisms must evolve from simple Logic Programming reasoning to Extend Logic Programming reasoning with incomplete information and, from here, to Extend Logic Programming reasoning with temporal and priority considerations. Delegation, gratitude and agreement are some of the most important characteristics to be taken into account when negotiating, these characteristics introduce non-linearities in the reasoning process in order to condition the behaviour of an agent. Once again, mathematical logic serves as a powerful tool to model such subjective situations.

---

<sup>1</sup>- Corresponding author. Tel.: +351 253604437; fax: +351 253604471; email: pjon@di.uminho.pt

---

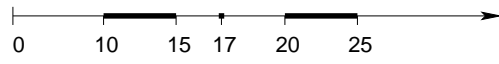
## References

- [1] C. Analide, J. Neves, Representação de Informação Incompleta, in: Proc. CAPSI, Conferência da Associação Portuguesa de Sistemas de Informação, Guimarães, Portugal, 2000.
- [2] C. Baral, M. Gelfond, Logic Programming and Knowledge Representation, Journal of Logic Programming, 19,20 (1994) 73 - 148.
- [3] L. Brito, P. Novais, J. Neves, Mediation, Agreement and Gratitude in Strategic Planning for Virtual Organisations, in: B. Sharp and J. Filipe and J. Cordeiro, ed., Enterprise Information Systems (Kluwer, 2001) 190-196.
- [4] L. Brito, P. Novais, J. Neves, On the Logical Aspects of Argument-based Negotiation among Agents, in: Proc. CIA 2001 - Fifth Int. Workshop on Cooperative Information Agents, Lectures Notes in Artificial Inteligence, vol. 2182 (Springer, Berlim, 2001) 178 - 189.
- [5] L. Brito, P. Novais, J. Neves, Temporality, Priorities and Delegation in an E-Commerce Environment, in: Proc. 14th Bled Electronic Commerce Conference, Slovenia (2001) 53 - 68.
- [6] R. Kowalski, F. Toni, Abstract Argumentation, Artificial Intelligence and Law Journal 4 (3-4), special issue on Logical Models of Argumentation, (Kluwer, 1996) 275 - 296.
- [7] R. Kowalski, F. Toni, Argument and Reconciliation, in: Proc. Int. Symposuim on Fifth Generation Computer Systems '94, Workshop on Legal Reasoning, Tokyo (1994) 9 - 16.

- [8] J. McCarthy, Programs with Common Sense, in: Proc. Teddington Conference on the Mechanization of Thought Processes, London (1959) 75 - 91.
- [9] T. Norman, C. Sierra, N. Jennings, Rights and Commitment in Multi-Agent Agreements, in: Proc. ICMAS 98 - Int. Conf. on Multi-Agent Systems, France, (1998) 222 - 229.
- [10] J. Neves, A Logic Interpreter to handle time and negation in logic data bases, in: Proc. ACM 1984 Annual Conference, San Francisco, California, USA, (1984) 50 - 54.
- [11] P. Novais, L. Brito, J. Neves, Experience-Based Mediator Agents as the Basis of an Electronic Commerce System, in: Proc. Workshop 2000 - Agent-Based Simulation, Passau, Germany, (SCS, 2000) 151 - 156.
- [12] P. Novais, J. Neves, L. Brito, J. Machado, Argumentative Procedures in e-Commerce Environments, in: Proc. IFIP - 2th Conf. on e-Commerce, e-Business, e-Government (Kulwer Academic Pub., 2002) 703 - 716.
- [13] H. Prakken, Logical Tools for Modelling Legal Argument, Doctoral Dissertation, Free University, Amsterdam, 1993.
- [14] S. Russel, P. Norvig, Artificial Intelligence - A Modern Approach (Prentice-Hall, Inc., New Jersey, USA, 1995).
- [15] G. Sartor, A Simple Computational Model for Nonmonotonic and Adversarial Legal Reasoning, in: Proc. 4th Int. Conf. on Artificial Intelligence and Law (ACM, 1993) 192 - 201.

## List of Figures

Figure 1: A right-to-deal clause with temporal considerations



`right_to_deal(p1,[c1,c2],john):[[10,15],[17,17],[20,25]].`

**Figure 1:** Knowledge -Based Systems; Paulo Novais, Luís Brito and José Neves