Properties and Complexity in Feasible Logic-Based Argumentation for Electronic Commerce

Luís Brito, José Neves

Departamento de Informática
Escola de Engenharia
Universidade do Minho
Campus de Gualtar
4710-057 Braga
PORTUGAL
Phone: +351-253-604470 Fax: +351-253-604471
email: {lbrito,jneves}@di.uminho.pt
http://alfa.di.uminho.pt/~lbrito

Submission for: PAPER TRACK

Abstract. Logic-Based Argumentation (LBA) exhibits unique properties and advantages over other kinds of argumentation procedures, amongst them: the adequacy to logic-based pre-argument reasoning, similarity to the human reasoning process, reasoning with incomplete information and argument composition and extension. Logic enables a formal specification to be built and a quick prototype to be developed. In order for LBA to achieve feasibility in Electronic Commerce (EC) scenarios, a set of properties must be present: self-support, correctness, conjugation, temporal containment and acyclicity. At the same time, LBA is shown to achieve stability in argument exchange (*guaranteed success problem*) and, depending on the definition of *success*, computational efficiency at each round (*success problem*).

Keywords: logic-based argumentation, electronic commerce, success problem, guaranteed success problem

Conference Topics:

1. AI Foundations and Knowledge Representation;
2. Knowledge Engineering (Ontologies, Knowledge Acquisition, Knowledge Representation, Reuse of Knowledge);
3. Reasoning Models: Non-monotonic Reasoning;
4. Multi-Agent Systems and Distributed AI.

Properties and Complexity in Feasible Logic-Based Argumentation for Electronic Commerce

Luís Brito, José Neves

Departamento de Informática
Universidade do Minho
Braga
PORTUGAL
{lbrito,jneves}@di.uminho.pt
http://alfa.di.uminho.pt/~lbrito

Abstract Logic-Based Argumentation (LBA) exhibits unique properties and advantages over other kinds of argumentation procedures, amongst them: the adequacy to logic-based pre-argument reasoning, similarity to the human reasoning process, reasoning with incomplete information and argument composition and extension. Logic enables a formal specification to be built and a quick prototype to be developed. In order for LBA to achieve feasibility in Electronic Commerce (EC) scenarios, a set of properties must be present: self-support, correctness, conjugation, temporal containment and acyclicity. At the same time, LBA is shown to achieve stability in argument exchange (*guaranteed success problem*) and, depending on the definition of *success*, computational efficiency at each round (*success problem*).

Keywords: logic-based argumentation, electronic commerce, success problem, guaranteed success problem

1 Introduction

The use of logic (be it propositional, extended logic programming, modal or any other) enables systems to be modeled with the added benefit of mathematical notation and non-ambiguity. Logic programming tools even provide a working prototype for the modeled system, amazingly reducing the time between formal specification and prototype development/testing. Argumentation systems benefit from the use of logic for two reasons: the intrinsic logic behind argumentation [1,7] and the already stated benefits in the software development cycle.

Electronic Commerce (EC) environments provide an unparalleled arena for the combination of logic and argumentation [5]. Logic provides the formal tools for the sound development of agents and agent reasoning mechanisms. Argumentation provides a way to exchange justified information among business counterparts (be it in Business-to-Consumer – B2C – or Business-to-Business – B2B – scenarios) or even to develop negotiation techniques that aim at shorter times for each deal (with more information present at each stage) [11,16]. However, the feasibility of Logic-Based Argumentation (LBA) for EC can only be determined if two problems are approached:

- **EC-directed properties**: in order for LBA to be feasible for EC, arguments must exhibit properties that reduce algorithmic complexity, guarantee acyclicity and deliver correction, to name a few;
- **complexity of success and guaranteed success problems**: once the desired kind of arguments is chosen and an algorithm determined, feasibility can only be achieved by argumentation procedures which enable success determination and guarantee success with a relatively low complexity.

On section 2 a formalization for LBA is presented, together with advantages, mathematical foundations and the proof of each necessary property for EC feasibility. On section 3 the success and guaranteed success problems are stated and the complexity for LBA is presented. Finally, sections 4 and 5 show related work and some conclusions.

The main contributions of this work are: (i) deliver the power of mathematical logic to argumentative procedures in EC; (ii) state the main advantages of LBA; (iii) state and prove some of the most important properties for feasible argumentation; (iv) determine if LBA provides success determination and success guarantees.

2 Logic-Based Argumentation for Electronic Commerce

2.1 Advantages

Although the use of logic has been questioned in the field of argumentation [12], logic-based argumentation still presents a set of characteristics which can not be measured by a simplistic computational efficiency metric, such as [5,6]:

- **adequacy to logic-based approaches to pre-argument reasoning**: some agent development strategies [8,15] define a stage that precedes the instant an agent starts to articulate an argument. This stage is called pre-argument reasoning and enables the agent to reason about such things as the *right to deal* some product or the *right to deal* with some counterpart. Due to the fundamental use of logic as a formalization tool and the manipulation of a logic Knowledge Base (KB) [8] a set of rules is available in order for an argument to be formulated;
- **similarity to the human reasoning processes**: the use of logical mechanisms in reasoning and, in special, such inference mechanisms as *modus ponens*, enable easy construction of rules even by non-experts. On the other hand, the set of available rules (in an agent's KB) is largely human-readable;
- **reasoning with incomplete information**: the use of null values [2,7], in combination with negation as failure, enables the use of incomplete information and a reasoning mechanism that deals with uncertainty (i.e., the *unknown* valuation in clauses). An agent is able to construct arguments where some of the steps are not taken as simple *true* or *false* elements;

– **argument composition and extension**: the set of logical elements (rules) which compose an argument may be extended in order to strengthen the argument conclusion, therefore inumerous compositions might be available, which permits easy adaption to the specific kind of argument intention (e.g. information exchange). On the other hand, taking an argument for A and the insertion of a rule such as $B \leftarrow A$, an argument for B is trivially reached;

2.2 Mathematical Foundations

Extended Logic Programming (ELP) is a useful, simple and powerful tool for logic problems. If argumentation in EC-oriented agents is to be addressed through ELP the structure of each agent's KB needs to be defined. This KB is considered to be a collection of organized clauses (logic theory OT') that enable inferencing and, therefore, action justification and argument construction. Formalizing the definition of *knowledge clause*:

Definition 1. (knowledge clause) *The knowledge available in each agent is composed of logic clauses of the form* $r_k : P_{i+j+1} \leftarrow P_1 \wedge P_2 \wedge ... \wedge P_{i-1} \wedge not\, P_i \wedge ... \wedge not\, P_{i+j}.$*, where* $i, j, k \in N_0$*,* $P_1, ..., P_{i+j+1}$ *are literals; i.e, formulas of the form* p *or* $\neg p$*, where* p *is an atom. In these clauses* r_k*,* not*,* P_{i+j+1}*, and* $P_1 \wedge P_2 \wedge ... \wedge P_{i-1} \wedge not\, P_i \wedge ... \wedge not\, P_{i+j}$ *stand, respectively, for the clause's identifier, the* ***negation-as-failure*** *operator, the rule's consequent, and the rule's antecedent. If* $i = j = 0$ *the clause (rule) is called* ***fact*** *and represented as* $r_k : P_1.$ *.*

An ELP program (Π_{ELP}) is seen as a set of knowledge clauses as the ones presented above. Arguments are to be constructed from inference sequences over an agent's KB (in fact, a Π_{ELP}). The use of ELP in the KB enables a three-valued logic [4,2,8] which leads to the possibility of using *null values* to represent incomplete information. These *null values* combined with a meta-theorem solver enable the construction of arguments that rely not only on rules that are *positively triggered* (i.e., their logical valuation after variable instantiation is *true*) but on all the three logical valuations: *true*, *false* and *unknown*. This reasoning over incomplete and unknown information is extremely important in EC scenarios due to the pervasive nature of fuzzy negotiation situations (e.g. agent A is able to deal product P with agent B using the set of conditions C_1, however it is not known if it can do the same thing with a set C_2 – leading to further dialog). Formalizing the definition of *negotiation argument*:

Definition 2. (negotiation argument) *Taking ordered theory* OT'*, a negotiation argument is a finite, non-empty sequence of rules* $\langle r_1, ..., demo(r_i, V_i), ..., r_n \rangle$ *such that, for each sequence rule* r_j *with* P *as a part of the antecedent, there is a sequence rule* r_i $(i < j)$ *on which the consequent is* P *(*$demo(r_i, V_i)$ *represents the meta-theorem solver application over rule* r_i *and valuation* V_i*).*

The conclusion of an argument relates to the consequent of the last rule used in that same argument. Therefore, having in mind the use of such premise in further definitions, a formal statement of argument conclusion is to be reached.

Definition 3. (conclusion) *The conclusion of an argument* $A_1 = \langle r_1, ..., r_n \rangle$, $conc(A_1)$*, is the consequent of the last rule* (r_n) *an none other than that one.*

Notice that, through the current definition of *negotiation argument*, it is possible to build *incoherent arguments*; i.e., it is possible to build arguments where there are rules that attack (deny) previous rules stated in the sequence. The formal definition of coherency is provided by an invariant definition:

Definition 4. (coherency) *An argument* $A_1 = \langle r_1, ..., r_n \rangle$ *is said to be "coherent" iff* $\neg\exists_{a_i,a_j} a_i, a_j \in subarguments(A) \wedge i \neq j : a_i\, attacks\, a_j$.

Taking into account the two forms of argument attack (*conclusion denial* and *premise denial*), a *conflict* among two opposing agents (e.g., buyer/seller) can be formally specified.

Definition 5. (conflict/attack over negotiation arguments)
Let $A_1 = \langle r_{1,1}, ..., r_{1,n} \rangle$ *be the argument of agent 1 and* $A_2 = \langle r_{2,1}, ..., r_{2,m} \rangle$ *be the argument of agent 2. Then,*

(1) A_1 *attacks* A_2 *iff* A_1 *executes a conclusion denial attack or a premise denial attack over* A_2*;*

(2) A_1 *executes a conclusion denial attack over* A_2 *iff and* $conc(A_1) = \neg conc(A_2)$*;*

(3) A_1 *executes a premise denial attack over* A_2 *iff* $\exists r_{2.j} \in A_2 - conc(A_2) : conc(A_1) = \neg r_{2,j}$.

Once coherent arguments are exchanged, a negotiation history (set of exchanged arguments) of the issuer agent to the receiver agent can be defined.

Definition 6. (history) *The argumentation history of agent A to agent B is* $h_{A\rightarrow B} = \langle A_{1,A\rightarrow B}, A_{2,A\rightarrow B}, ..., A_{n,A\rightarrow B} \rangle$.

It is also important to state that *argumentation procedures* (i.e., the exchange of arguments amongst agents on a particular issue) should exhibit acyclicity. Once some conclusion has been put forward by one of agents and attacked by a counterpart, that same conclusion can not be put forward and attacked once again due to the danger of an *argumentative cycle*.

Definition 7. (termination) *Given the argumentation histories* $h_{A\rightarrow B}$, $h_{B\rightarrow A}$ *and arguments* $A_i \in h_{A\rightarrow B}$ *and* $A_j \in h_{B\rightarrow A}$, $\exists^1_{(i,j)}\, conc(A_i) = P \bigwedge conc(A_j) = \neg P$ *in order to deliver argumentative acyclicity and termination.*

2.3 Properties

After stating the mathematical foundations of LBA, it is now possible to present and prove a set of theorems that establish its most important features. By proving the validity of such properties, it is possible to ensure that EC-directed arguments based on the present LBA system ensure correction and feasibility.

EC needs: arguments which are supported and inferred from the specific knowledge of each agent (self-support property), truthful agents (non-contradiction property), easy to combine arguments (conjugation property), non-monotonous knowledge bases to capture the commercial reality (temporal containment property) and an acyclic line of argument (acyclicity property).

In order to prove many of these properties it must be ensured that *stable time intervals* are considered, once if long range reasoning is assumed and given the non-monotonous characteristics of each agent's KB (necessary to faithfully express real-world commerce) contradictory arguments might be generated. The arguments in an LBA system for EC exhibit, therefore, the following set of properties:

Property 1. ***(self-support)*** Given a stable time interval, argument $A_1 \in h_{A \to B}$ and having $A_1 \vdash P$, then $KB_A \vdash P$.

Proof. By $A_1 \in h_{A \to B}$ and the definition of *argument*, $A_1 = \langle r_1, ..., r_n \rangle$ being $r_i \in KB_A$. Therefore, $KB_A \supseteq A_1 \vdash P$ and by consequence $KB_A \vdash P$.

It is then proved that agent A can only support its arguments by the knowledge present at its own KB. □

Corollary 1. *Given a stable time interval,* $A_1 \in h_{A \to B}$ *and having* $conc(A_1) = P$*, then* $KB_A \vdash P$*.*

Property 2. ***(correctness or non-contradiction)*** Given a stable time interval, arguments $A_1 \in h_{B \to A}$, $A_2 \in h_{A \to B}$, $A_3 \in h_{A \to C}$, $A_4 \in h_{C \to A}$, argument conclusions $conc(A_1) = \neg P$, $conc(A_2) = P$, $conc(A_3) = P$ and an attack of A_4 over A_3, then $conc(A_4) \neq \neg P$.

Proof. If an attack of A_4 over A_3 takes place, either the conclusion or the premises of A_3 are denied by A_4. Assume each situation separately:

- *conclusion attack*: $conc(A_4) = \neg conc(A_3) = \neg P$. Taking into account that $A_4 \subseteq KB_A$, then $KB_A \vdash \neg P$. However, the KB of each agent is coherent and by $conc(A_2) = P$ it is know that $KB_A \vdash P$, therefore it must happen $conc(A_4) \neq \neg P$;
- *premise attack*: $conc(A_4) = \neg Q$ $(Q \in A_3 - [conc(A_3)])$. By the definition of *argument conclusion* and by $conc(A_3) = P$ it is known that $conc(A_4) \neq \neg P$.

It is then proved that agent A is unable to "lie" (i.e., state $\neg P$ after having stated P in the same time frame). □

Property 3. ***(conjugation)*** Given a stable time interval, argument $A_1 \in h_{A \to B}$, argument $A_2 \in h_{A \to B}$, $conc(A_1) = P_1$, $conc(A_2) = P_2$ and a general rule $Q \leftarrow P_1, P_2.$, then argument $A = A_1 \odot A_2 \odot Q \leftarrow P_1, P_2.$ (where $\odot$ stands as the concatenation operator) delivers $conc(A) = Q$.

Proof. Taking $conc(A_1) = P_1$ and $conc(A_2) = P_2$, then by the definition of *argument* and *argument conclusion* argument A is valid only if $A' = \langle P_1, P_2, Q \leftarrow P_1, P_2. \rangle$

is valid (due to the fact that the justifications for A_1 and A_2 are independent). Once again, by the definition of *argument,* A' is valid and, by the definition of *argument conclusion,* $conc(A) = Q$.

It is then proved that agent A is able to compose valid arguments into new arguments with combined conclusions. □

Property 4. ***(temporal containment)*** Given time instants $t_1 \neq t_2$, KB_{t_1}(the agent's KB at time instant t_1), KB_{t_2} (the agent's KB at time instant t_2), argument $A = \langle r_1, ..., r_n \rangle$, $r_i \in KB_{t_1}$ and $conc(A) = P$, then *it can not* be concluded that $KB_{t_2} \vdash P$.

Proof. Proceeding by an absurd reduction proof, assume that with time instants $t_1 \neq t_2$, KB_{t_1}, KB_{t_2}, argument $A = \langle r_1, ..., r_n \rangle$, $r_i \in KB_{t_1}$ and $conc(A) = P$, then *it can* be concluded that $KB_{t_2} \vdash P$. By having, for example, $KB_{t_1} = \{a.; b \leftarrow a, c.\}$, $KB_{t_2} = \{\neg a.\}$, $t_1 \neq t_2$ and $A = \langle a \rangle$, it can be concluded that $conc(A) = a$ and, by the taken assumption, $KB_{t_2} \vdash a$, however by looking at KB_{t_2} is is easily seen that $KB_{t_2} \nvdash a$. The initial assumption is, therefore, false.

It is then proved that the fact that an argument is generated at a given time instant, does not mean that the KB of that same agent is able to generate that same argument at a different time instant. □

Property 5. ***(acyclicity)*** Given an acyclic KB, then $\forall A, A \in h_{A \rightarrow B}$ generated from KB is acyclic.

Proof. Take the Atom Dependency Graphs [4] ADG_A and ADG_{KB} (i.e., a graph which has ground atoms at each vertex and directed edges labeled with $\langle P_i, P_j, s \rangle$, representing the existence of a rule with P_i at the head, P_j in the body and $s \in \{+, -\}$ if the atom is positive or negative, respectively) derived from the ELP programs present at A and KB, respectively. Having $A \subseteq KB$, if $\forall_i (u_i, u_{i+1}) \in edges(ADG_A)$ (with $u_i \in vertices(ADG_A)$) then $(u_i, u_{i+1}) \in edges(ADG_{KB})$ (with $u_i \in vertices(ADG_{KB})$). If a cycle exists within ADG_A, $\exists_i (u_i, u_i) \in edges(ADG_A)$ and due to $A \subseteq KB$, $\exists_i (u_i, u_i) \in edges(ADG_{KB})$, once ADG_{KB} is acyclic a cycle in ADG_A can not exist.

It is then proved that an acyclic ELP KB (which implies a *terminating program* [4,3]) delivers acyclic (therefore, *terminating*) arguments. □

3 The Success and Guaranteed Success Problems

3.1 Introduction

Although the presented properties show that LBA has the necessary semantic and syntactic soundness to be used in EC environments, it must be proved that computational feasibility is also present. It must be shown that LBA protocols exhibit computational feasibility at each round and, at the same time, stability or termination is reached. These problems are also known as the *success* and *guaranteed success* problems, respectively.

In a negotiation process, through which the acceptability region of each agent is constrained [11], there is an active adversarial process that proceeds by an argument/counter-argument mechanism, where an agent successively attacks another agent's premises or conclusions. On the other hand, there is a different situation where argumentation is used as a way of exchanging justified information and support the decision-making process. Therefore, the *success* and *guaranteed success* problems have to be considered for these two situations.

3.2 The Success Problem

In the case of EC-directed negotiation, success can be measured in many ways (e.g. the number of "victories" over a set of rounds, or the lowest argument length average), however in pragmatic terms victory rarely depends on a successful history of arguments (which serve in adversarial argumentation, as a way to constrain the acceptability space of each agent) but rather on the conclusions of the present round (which may be the last or just a step on the overall argumentation history). A definition for *success* is in order:

Definition 8. (success) *A set of exchanged LBA arguments* $\bigcup_{i,j \in Agents, i \neq j} h_{i \to j}$ *exhibits* ***success*** *if* $\bigwedge_{i,j \in Agents, i \neq j} conc_{ext}\left(A_{|h_{i \to j}|}\right) \nvdash \bot$*, where* $conc_{ext}()$*, Agents and* $A_{|h_{i \to j}|}$ *stand, respectively, for the* ***extended conclusion****, the set of involved agents and the last argument sent from agent* i *to agent* j*. The* ***extended conclusion*** *results from extending the conclusion function by assigning logic values to two specific situations: concession (an agent quits by agreeing with the counterparts' conclusions) and drop-out (an agent quits by refusing to agree with the counterparts' conclusions), formally:*

$$conc_{ext}(A) = \begin{cases} \top, & if\, concession \\ \bot, & if\, drop-out \\ conc(A), & otherwise \end{cases}.$$

Considering that each conclusion is a ground literal, the previous definition of *success* leads to an algorithm that searches for the presence of $\bot$ or the presence of both P and $\neg P$, which can easily be done in $O\left(|Agents|^2\right)$. However, if *success* is defined in terms of the total argument and not only its conclusion, by assuming propositional rules an equivalent to the *logic satisfiability problem* is achieved, which is proven to be NP-complete [21,20,10]. By assuming DATALOG restrictions, complexity is proven to be EXPTIME-complete [10].

In the case of a non-adversarial argument exchange, the existence of premise and conclusion-denial situations lead to a much easier to solve *success* problem. In an EC-directed environments where non-adversarial argumentation occurs (typically B2B situations), each agent presents (in an informative way) what it can deduce from its knowledge about the world, therefore $conc_{ext}()$ on *Definition 8* is to be changed to allow a permanent *success* situation:

$$conc_{ext}(A) = \top.$$

3.3 The Guaranteed Success Problem

Although it is important to determine the eventual *success* at each argumentation round, LBA for EC can only be considered feasible if the *guaranteed success problem* is not computationally intractable; i.e., it is possible to achieve stability on the set of exchanged arguments. The pragmatic characteristics of EC do not support large (or even infinite) argument exchange sets.

In the case of non-adversarial argumentation (e.g. many B2B situations), the necessity to reach a stable situation does not directly arise. Though each agent uses the premise and conclusion denial tools to generate counter-arguments, the aim of such action is not to arrive at an agreement on some knowledge but rather a cooperative information exchange [7]. Therefore, *success is guaranteed* at each round.

Adversarial argumentation is more complex than the non-adversarial one. By considering that each argument was built through the use of a language based on propositional Horn clauses $\mathcal{L}_0^{HC}$, it has been proven that the *guaranteed success problem* for such a language is co-NP-complete [21,20]. Nonetheless such an approach, the expressiveness from the desired ELP is reduced to propositional logic while maintaining an high algorithmic complexity. However, the set of properties present in LBA (and previously proven) yield an interesting conclusion:

Theorem 1. *Assuming that each round is a stable time interval, then adversarial LBA exhibits* ***guaranteed success.***

Proof. Given the stable time interval of a round, *Properties 2* (*non-contradiction*) and *5* (*acyclicity*), and *Definition 7* (*termination*), it is easy to conclude that each agent can not generate argumentative cycles through "lies" or arguments with cyclic clauses and, at the same time, it can not incur in the use of arguments which have been denied by a counterpart.

Being n_0 the finite size of the Atom Dependency Graph (i.e., the total number of edges and vertexes) associated with the knowledge of a given agent (ADG_{KB}), the argumentative process proceeds, round by round, on an acyclic argumentative path, progressively constraining the possible set of usable inferences to perform premise and conclusion-denial attacks. Therefore, ADG_{KB} has the number of traversable edges and vertexes, in order to reach some conclusion P, progressively reduced:

$$n_0, n_1, n_2, ..., n_m, \;\; n_0 > n_1 > n_2 > ... > n_m$$

where n_i is the size of the traversable ADG_{KB} at each round. It is then easy to see that, in a number of steps lower than n_0 (therefore finite), no path will be available on ADG_{KB} and a *concession* or *drop-out* occurs. □

4 Related Work

The study of argumentation through mathematical logic (especially ELP) goes back to work done by Loui [14], Simari, Chesñevar and García [18], and by

Kowalski and Toni [13]. Formalization through reasoning models happened even earlier by Toulmin [19] and, in philosophical terms, during Classic Antiquity.

The use of LBA in the Law arena, is present in the work of Prakken and Sartor [17]. However, the use in EC scenarios was proposed by Jennings, Parsons, Noriega and Sierra [11]. The formalization and viability study for B2C, B2B and cooperative argumentation was approached by Brito and Neves [5,6] combined with a presentation of a 4-step approach to agent development by Brito, Novais and Neves [7,9].

A complexity study on the use of logic in negotiation has been presented by Wooldridge and Parsons [21,20].

5 Conclusions

The use of ELP is important either to formalize the reasoning mechanisms and the knowledge exchange in EC scenarios, or to quickly develop a working prototype. LBA exhibits a set of characteristics which are unique to this kind of argumentation such as: dealing with incomplete information and similarity to the human reasoning process.

Formalizing adversarial or even cooperative argument exchanges in EC demands an important set of characteristics to be present on LBA, which are: self-support, correctness, conjugation, temporal containment and acyclicity. It has been proven that, through the present LBA formalization, the necessary properties are available.

However it is necessary to evaluate, at each round, the *success* of an argumentative procedure and it must be proven that stability is reached at some stage (*guaranteed success*). These questions are extremely important in EC scenarios and condition the feasibility and viability of LBA for EC. It has been proven that for non-adversarial situations the *success* and *guaranteed success* problems are trivially solved. As for adversarial situations, the *success* problem can be reduced to polynomial complexity (depending on the definition of *success*) and there is *guaranteed success*.

References

1. L. Amgoud, N. Maudet, and S. Parsons. Modelling Dialogues using Argumentation. In *Proceedings of the 4th International Conference on Multi-Agent Systems*, Boston, USA, 2000.
2. C. Analide and J. Neves. Representation of Incomplete Information. In *Conferencia da Associacao Portuguesa de Sistemas de Informaçao (CAPSI)*, Universidade do Minho, Guimaraes, Portugal, 2000.
3. K. Apt and M. Bezem. Acyclic Programs. *New Generation Computing*, 9(3,4):335–365, 1991.
4. C. Baral and M. Gelfond. Logic Programming and Knowledge Representation. Technical report, Computer Science Department, University of Texas at El Paso, El Paso, Texas, USA, 1994.

5. L. Brito and J. Neves. Discussing Logic-Based Argumentation in Extended Electronic Commerce Environments. In *Workshop "Adventures in Argumentation" at the 6th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'01)*, Toulouse, France, September 2001.
6. L. Brito and J. Neves. Argument Exchange in Heterogeneous Electronic Commerce Environments. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems AAMAS 2002*, Bologna, Italy, June 2002. ACM Press.
7. L. Brito, P. Novais, and J. Neves. On the Logical Aspects of Argument-based Negotiation among Agents. In Matthias Klush and Franco Zambonelli, editors, *Cooperative Information Agents V, Proceedings of the 5th International Workshop CIA 2001*, number 2182 in Lecture Notes in Artificial Intelligence, pages 178–189, Modena, Italy, September 2001. Springer-Verlag: Heidelberg, Germany.
8. L. Brito, P. Novais, and J. Neves. Temporality, Priorities and Delegation in an E-Commerce Environment. In *Proceedings of the 14th Bled Electronic Commerce Conference (Bled01)*, Bled, Slovenia, July 2001.
9. L. Brito, P. Novais, and J. Neves. *Intelligent Agent Software Engineering*, chapter The Logic Behing Negotiation: From Pre-Argument Reasoning to Argument-Based Negotiation. IDEA Group, 2002.
10. E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and Expressive Power of Logic Programming. *ACM Computing Surveys*, 3(33):374–425, 2001.
11. N. Jennings, S. Parsons, P. Noriega, and C. Sierra. On Argumentation-based Negotiation. In *Proceedings of the International Workshop on Multi-Agent Systems*, Boston, USA, 1998.
12. H. Jung, H. Tambe, and S. Kulkarni. Argumentation as Distributed Constraint Satisfaction: Applications and Results. In *Proceedings of the International Conference on Autonomous Agents (Agents01)*, Montreal, Canada, 2001.
13. R. Kowalski and F. Toni. Argument and Reconciliation. In *Proceedings of the International Symposium on the Fifth Generation Computer Systems: Workshop on Legal Reasoning*, pages 9–16, Tokyo, Japan, 1994.
14. R. Loui. Defeat Among Arguments: a System of Defeasible Inference. *Computational Intelligence*, (3):100–106, 1987.
15. P. Novais, L. Brito, and J. Neves. Developing Agents for Electronic Commerce: A Constructive Approach. In *Proceedings of the World Conference on Systemics, Cybernetics and Informatics (SCI2001)*, Orlando, Florida, 2001.
16. S. Parsons and N. Jennings. Argumentation and Multi-Agent Decision Making. In *Proceedings of the AAAI Spring Symposium on Interactive and Mixed-Initiative Decision Making*, pages 89–91, Stanford, USA, 1998.
17. H. Prakken and G. Sartor. Argument-based Logic Programming with Defeasible Priorities. *Journal of Applied Non-Classical Logics*, (7):25–75, 1997.
18. G. Simari, C. Chesñevar, and A. García. Modelling Argumentation in a Logic Programming Setting: Formalization and Logic Properties. In *Workshop "Adventures in Argumentation" at the 6th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'01)*, Toulouse, France, September 2001.
19. S. Toulmin. *The Uses of Arguments*. Cambridge University Press, 1958.
20. M. Wooldridge and S. Parsons. Languages for Negotiation. In *Proceedings of 14th European Conference on Artificial Intelligence (ECAI2000)*, 2000.
21. M. Wooldridge and S. Parsons. On the Use of Logic in Negotiation. In *Proceedings of the Autonomous Agents 2000 Workshop on Agent Communication Languages and Conversation Protocols*, Barcelona, Spain, 2000.