

An Overview of Argumentation Semantics

Una Revisión de las Semánticas de Argumentación

Juan Carlos Nieves¹, Mauricio Osorio² and Ulises Cortés¹

¹Universitat Politècnica de Catalunya

Software Department (LSI)

c/Jordi Girona 1-3, E08034, Barcelona, Spain

Email: jcnieves@lsi.upc.edu, ia@lsi.upc.edu

²Universidad de las Américas - Puebla

CENTIA, Sta. Catarina Mártir

Cholula, Puebla, 72820 México

Email: osoriomauroi@googlegmail.com

Article received on April 14, 2008; accepted on June 20, 2008

Abstract

The main purpose of argumentation theory is to study the fundamental mechanisms that humans use in argumentation, and to explore ways to implement these mechanisms on computers. During the last years, argumentation has been gaining increasing importance in Computer Science, especially in areas as Artificial Intelligence, e-commerce, Multi-agent Systems and Decision-Making.

In this paper, we present a brief overview of abstract argumentation semantics. In order to promote and disseminate this young area, we describe the fundamental role of argumentation in a medical application. Moreover, we present some results in order to close the huge gap between argumentation theory and argumentation systems. We will see that these results also suggest a general method for exploring some challenges in argumentation theory.

Keywords: Argumentation Theory, Logic Programming, Non-Monotonic Reasoning.

Resumen

El principal propósito de la teoría de argumentación es el estudio de los mecanismos básicos que los humanos usan en argumentación y además explorar métodos para implementar dichos mecanismos en las computadoras. Durante los últimos años, argumentación ha ganado importancia en el área de las ciencias de la computación muy en especial en los campos de la inteligencia artificial, comercio electrónico, sistemas multi - agentes y toma de decisiones.

En este artículo, presentamos una breve revisión de los patrones más aceptados en la selección de argumentos --- a dichos patrones se les llama semánticas de argumentación. Con el propósito de promover y difundir esta joven área de investigación, se describirá el uso de argumentación en una aplicación médica. Además, presentaremos algunos resultados que contribuyen a la integración de modelos teóricos de argumentación a sistemas reales basados en argumentación.

Palabras Clave: Teoría de Argumentación, Programación Lógica, Razonamiento No Monótono.

1 Introduction

Argumentation theory, or argumentation, embraces the arts and sciences of civil debate, dialogue, conversation, and persuasion. It studies rules of inference, logic, and procedural rules in both artificial and real world settings. Argumentation is concerned primarily with reaching conclusions through logical reasoning, that is, claims based on premises. Although including debate and negotiation which are concerned with reaching mutually acceptable conclusions, argumentation theory also encompasses eristic dialog, the branch of social debate in which victory over an opponent is the primary goal. This art and science is often the means by which people protect their beliefs or self-interests in rational dialogue, in common parlance, and during the process of arguing.

Argumentation is also a formal discipline within Artificial Intelligence (AI) where the aim is to make a computer assist in or perform the act of argumentation. In fact, during the last years, argumentation has been gaining increasing importance in Multi-Agent Systems (MAS), mainly as a vehicle for facilitating *rational interaction* (*i. e.*

interaction which involves the giving and receiving of reasons). A single agent may also use argumentation techniques to perform its individual reasoning because it needs to make decisions under complex preferences policies, in a highly dynamic environment.

Argumentation theory is also regarded as an approach of non-monotonic reasoning since it formalizes non-monotonic reasoning as the construction and comparison of arguments for and against certain conclusions. Non-monotonicity arises from the fact that new information may give rise to new counterarguments that defeat the original argument. In general, three ways of defeat are distinguished: arguing for a contradictory conclusion (rebutting), arguing that an inference is incorrect (undercutting), or denying a premise (premise-attack); in all three cases also considerations of strength of preference can be involved. In fact, most existing argumentation models allow for only one or two of the kinds of defeat. Inference in argumentation models is defined relative to a set of arguments and a binary attack relation between them. Typically, they classify arguments in three classes: the *acceptable* or justified arguments, the *defeated* or overruled arguments, and the *ties*, *i.e.*, the arguments, which are involved in an irresolvable conflict. Corresponding notions of propositional inference can be defined in terms of the status of arguments of which they are conclusions (the interested reader can find in, (Prakken and Vreeswijk 2002; Chesñevar et al. 2000), a good introduction to argumentation theory; moreover in, (Parsons et al. 2003; Reed and Norman 2004; Prakken 2005; Caminada and Amgoud 2007; Bench-Capon and Dunne 2007; Rahwan and McBurney 2007; Hitchcock and Verheij 2007; Alsinet et al. 2008; Nieves et al. 2008) the reader can find the some recent results and challenges *w. r. t.* argumentation theory in general)

Argumentation models are a particular group of patterns of inference, where arguments for and against a certain claim are produced and evaluated, to test the tenability of the claim, (Prakken and Vreeswijk 2002). In general, an argumentation model contains five elements:

1. an underlying logical language;
2. a definition of an argument (usually it is a proof of the underlying logic or a set of premises of such a proof);
3. definitions of conflicts between arguments and of attack among arguments;
4. a preference over claims (sentences in the object language) and an induced preference over arguments (not all systems have it);
5. a definition of the status of arguments.

The statuses of arguments are obtained by analyzing the attack relationship among arguments. There are two status-assignment approaches: a unique-status-assignment approach and a multiple-status-assignment approach.

Although several approaches have been proposed for capturing representative patterns of inference in argumentation theory, Dung's approach, presented in (Dung 1995), is a unifying framework which has played an influential role on argumentation research and AI. In fact the model suggested by Dung has given rise to an extensive body research with particular concentration on the following (Bench-Capon and Dunne 2007):

- Extension based semantics of argumentation.
- Algorithmic and complexity issues in argumentation.
- Dialogue processes for deciding acceptability.

Dung's approach is regarded as an abstract model where the main concern is to find the set of arguments which are considered as acceptable *i. e.*, to find sets of arguments which represent coherent points of view. The strategy for analyzing the attack relationships, and then inferring the sets of acceptable arguments, is based on extension based semantics. The kernel of Dung's framework is supported by four extension based semantics (some times we will refer to them also as abstract argumentation semantics): *grounded semantics*, *stable semantics*, *preferred semantics*, and *complete semantics*. The grounded semantics represents a unique-status-assignment approach and the other ones represent multiple-status-assignment approach. Although each abstract argumentation semantics represent a different

pattern of inference in argumentation theory, all of them have as common point the concept of *admissible set*. An admissible set represents a coherent point of view in a conflict of arguments.

To regard arguments as abstract concepts, as it is done in Dung's approach (Dung 1995), is a powerful tool which affords a formalism that focuses on the relationship between individual arguments as a means of defining diverse ideas of acceptance. Some authors point out that *the preferred semantics* is the particular interest since this represents maximal coherent point positions that can be defended against all attackers (Dunne and Bench-Capon 2004; Baroni et al. 2005; Bench-Capon and Dunne 2007).

Even though Dung's approach is a versatile and powerful tool for the abstract analysis of defeasible reasoning, one can find some potential problems. For instance, even though every argumentation framework has some preferred extension, this may simply be the empty set of arguments. Although this problem is avoided by the stable semantics, this semantics has the problem that there are argumentation frameworks which have no stable extensions. The grounded semantics does not have the problem of the stable semantics since it is always defined; however, the grounded extension can be simply the empty set of arguments as the preferred semantics.

Another important concern in abstract argumentation theory is the *computational complexity* of the decision problems that has been shown to range from linear to $\Pi_2^{(p)}$ -complete. A summary of this is given in Table 1 (practically all this table was taken from (Dunne 2007)).

As we can see, the computational complexity of the decision problem of argumentation semantics as the preferred is hard. However, recognizing the benefits of Dung's approach, a number of algorithms has been proposed in the literature (Besnard and Doutre 2004; Cayrol et al. 2003; Dung et al. 2006; Dung et al. 2007; Doutre and Mengin 2001; Nielsen and Parsons 2006; Egly and Woltran 2006).

Table 1. Decision problems in finite argumentation frameworks

	Instance	Decision Question	Complexity
(a)	$AF = \langle AR, attacks \rangle, x \in AR$	Is x in the <i>grounded</i> extension of AF ?	linear
(b)	$AF = \langle AR, attacks \rangle, x \in AR$	Is x in any <i>preferred</i> extension?	NP-complete
(c)	$AF = \langle AR, attacks \rangle, x \in AR$	Is x in any <i>stable</i> extension?	NP-complete
(d)	$AF = \langle AR, attacks \rangle$	Does AF have a <i>non-empty</i> preferred extension?	NP-complete
(e)	$AF = \langle AR, attacks \rangle$	Does AF have any <i>stable</i> extension?	NP-complete
(f)	$AF = \langle AR, attacks \rangle, x \in AR$	Is x in <i>every</i> stable extension?	CO-NP-Complete
(g)	$AF = \langle AR, attacks \rangle, x \in AR$	Is x in <i>every</i> preferred extension?	$\Pi_2^{(p)}$ -complete
(h)	$AF = \langle AR, attacks \rangle$	Is AF a coherent argumentation framework?	$\Pi_2^{(p)}$ -complete

The importance of finding efficient and practical methods for implementing argumentation technology is one of the priorities of the argumentation research community. For instance, one of the two main objectives of the European Project ASPIC¹ was

To develop efficient proof procedures and software component implementations of these models for deployment in real-world applications.

The diversity of the techniques used in the algorithms for inferring argumentation semantics is quite wide, one can find algorithms based on enumerative techniques, (Doutre and Mengin 2001; Nielsen and Parsons 2006), dialectic procedures, (Dung et al. 2006; Dung et al. 2007), model checking, (Besnard and Doutre 2004; Nieves et al. 2005b; Egly and Woltran 2006; Nieves et al. 2008). It is worth to comment that there are some interesting complexity-theoretic analyses as the presented in (Dunne 2002) that indicate that a number of computational questions remain difficult. However, Dunne leaves open the possibility that propositional formulae offer concise encodings for inferring argumentation semantics as the preferred semantics.

¹ ASPIC <http://www.argumentation.org/> --- Argument Service Platform With Integrated Components.

In this paper, we will summarize some key important encodings in order to study argumentation semantics. In particular, we concentrate our attention in some logic programming encodings which have been introduced in the last years. We will see that these encodings have important features in order to infer argumentation semantics as the preferred semantics; and even more, to perform some decision questions as the questions presented in Table 1. In fact, we will comment that some authors have used these encodings in order to define new argumentation semantics in terms of logic programming semantics.

Since one of the main objectives of this paper is to study, disseminate and promote argumentation technology, in the first part of this paper, we present a brief description of the argumentation approach based on extension based semantics and a real application of argumentation theory in a real domain. In particular, we describe the integration of argumentation in a multi-agent system (called CARREL) in order to support medical decision-making, (Vázquez-Salceda *et al.* 2003).

The rest of paper is divided as follows: In § 2, a concise description of Dung's approach is presented. In § 3, an application of argumentation is described. This application is in the context of multi-agent systems applied in health care. In § 4, a declarative problem solving approach is presented in order to infer admissible sets and preferred extensions. This approach is based on *enumerate* and *eliminate* approach. In §5, it is commented some features of a suitable codification in order to consider Dung's approach as logic programming. In § 6, some results of argumentation in terms of logic programming semantics are presented. We will see that these results suggest some mechanics in order to perform decision questions in finite argumentation frameworks (see Table 1). In § 7, it is described a proposal in term of logic programming semantics in order to overcome some challenges in argumentation theory. Finally in the last section, we present our conclusions.

2 Abstract Argumentation Theory

In order to have a clear presentation of this paper, we will make a small introduction to Dung's approach. This approach is based on simple concepts that can be understood with simple examples (see (Dung 1995) for more technical details).

The first concept that we will consider is the one of *argumentation framework*. An argumentation framework captures the relationships between the arguments (all the definitions of this subsection were taken from Dung's seminal paper (Dung 1995)).

Definition 1 An argumentation framework is a pair $AF = \langle AR, attacks \rangle$, where AR is a finite set of arguments, and $attacks$ is a binary relation on AR , *i. e.* $attacks \subseteq AR \times AR$.

For two arguments a and b , we say that a attacks b (or b is attacked by a) if $attacks(a, b)$ holds. Notice that the relation $attacks$ does not yet tell us with which arguments a dispute can be won; it only tells us the relation of two conflicting arguments. In order to illustrate this definition let us consider the following example which was taken from (Prakken and Vreeswijk 2002).

Example 1 Consider three arguments a , b and c such that a attacks b and b attacks c . A concrete version of this example is:

- c = Tweety flies because it is a bird.
- b = Tweety does not fly because it is a penguin.
- a = The observation that Tweety is a penguin is unreliable.

Notice that the argumentation framework which captures this example is $AF = \langle \{a, b, c\}, \{(a, b), (b, c)\} \rangle$. It is worth mentioning that any argumentation framework could be regarded as a directed graph. For instance, the graph representation of AF is presented in Fig. 1.



Fig. 1. Graph representation of the argumentation $AF = \langle \{a, b, c\}, \{(a, b), (b, c)\} \rangle$

Once a structure for capturing the conflicts that exist within a set of arguments is defined, now we require to define some minimal requirements to be satisfied within any computational sensible notion of “*collection of justified arguments*”.

Definition 2 A set S of arguments is said to be conflict-free if there are no arguments a, b in S such that a attacks b .

Conflict-freeness is a simple but important concept in abstract argumentation semantics. In fact, according to (Baroni and Giacomin 2007), conflict-freeness is viewed as a minimal requirement to be satisfied by any abstract argumentation semantics. However, conflict-freeness is a too weak condition in order to insure that a set of arguments S is collectively acceptable *e.g.* S could be attacked by arguments that do not belong to S . Hence, one requires adding conditions to a conflict-free subset of arguments in order to insure that a set of arguments is collectively acceptable. The definition of these conditions is the kernel of argumentation semantics, in fact some authors as (Bench-Capon and Dunne 2007) and Simari² have pointed out that the definition of these conditions is “*the plethora of argumentation semantics*”.

In this context, Dung defined the concept of *admissible set*. It captures *how* an argument that cannot defend itself can be protected by a set of arguments.

Definition 3 (1) An argument $a \in AR$ is *acceptable w. r. t.* a set S of arguments if and only if for each argument $b \in AR$: If b attacks a then b is attacked by an argument in S . (2) A conflict-free set of arguments S is *admissible* if and only if each argument in S is acceptable *w. r. t.* S .

Remark 1 We will say that any argument is defeated if and only if it is attacked by an acceptable argument.

The concept of admissible set is *the core* of the Dung's approach --- a good understanding of this concept will help to understand Dung's argumentation semantics and then this paper.

To illustrate Definition 3, let us consider Example 1. We can see that c is acceptable *w. r. t.* $\{a\}$, $\{a, b\}$, $\{a, c\}$ and $\{a, b, c\}$, but not *w. r. t.* $\{\}$ and $\{b\}$. Notice that $\{a, b, c\}$ and $\{a, b\}$ could not be admissible sets because they are not conflict-free sets. We can say that an admissible set represents a defendable point of view. For instance, in Example 1 there are three admissible sets: $\{\}$, $\{a\}$ and $\{a, c\}$. Intuitively, an admissible set is a coherent point of view. Since an argumentation framework could have several coherent points of view, one can take the maximum admissible sets in order to get maximum coherent points of view of an argumentation framework. This idea is captured by Dung's *preferred semantics*.

Definition 4 A preferred extension of an argumentation framework AF is a maximal (*w. r. t.* inclusion) admissible set of arguments of AF . The set of all the preferred extensions of AF is referred as the preferred semantics of AF .

Since an argumentation framework could have more than one preferred extension, the preferred semantics is called *credulous*. The argumentation framework of Fig. 1 has just one preferred extension which is $\{a, c\}$. Another credulous argumentation semantics introduced by (Dung 1995) is the so called *stable semantics*.

Definition 7 A conflict-free set of arguments A is called a stable extension if and only if S attacks each argument which does not belong to S .

² According to (Bench-Capon and Dunne 2007), this phase was done by Simari in the presentation of (Martinez *et al.* 2006) at COMMA 2006, 12th September 2006.

Dung showed that this semantics coincides with the notion of stable solutions of n-person games (Dung 1995). There is an interesting relationship between the stable semantics and the preferred semantics which is that every stable extension is a preferred extension, but not vice versa. Even though the stable semantics is closely related to a successful logic programming semantics with *negation as failure* called answer set semantics (Gelfond and Lifschitz 1988), it is often criticized because frequently this semantics is undefined (Caminada 2005, 2006).

Another important argumentation semantics introduced by Dung is *the grounded semantics*. The grounded semantics is able to capture some well-accepted argumentation approaches *e. g.*, (Prakken and Santor 1996) and (Simari and Loui 1992). This semantics is defined in terms of a *characteristic function*.

Definition 6 The characteristic function, denoted by F_{AF} , of an argumentation framework $AF = \langle AR, attacks \rangle$ is defined as follows:

$$\begin{aligned} F_{AF} &: 2^{AR} \rightarrow 2^{AR} \\ F_{AF}(S) &= \{a \mid a \text{ is acceptable w.r.t. } S\} \end{aligned}$$

Hence, by considering the characteristic function F_{AF} , the grounded semantics is defined as follows:

Definition 7 The grounded extension of an argumentation framework AF , denoted by GE_{AF} , is the least fixed point of F_{AF} .

The grounded semantics is regarded as a skeptical argumentation semantics. In fact any preferred extension is a superset of the grounded extension and the grounded extension is a subset of the skeptical version of the preferred semantics³. In (Dung 95), it was also showed that the grounded is the least (*w. r. t.* set inclusion) complete extension. The complete extensions define another credulous argumentation semantics introduced in (Dung 95).

In order to illustrate the definition, let us consider the argumentation framework of Fig. 1. Then

$$\begin{aligned} F_{AF}^0(\emptyset) &:= \{a\}, \\ F_{AF}^1(F_{AF}^0(\emptyset)) &:= \{a, c\}, \\ F_{AF}^2(F_{AF}^1(F_{AF}^0(\emptyset))) &:= \{a, c\}, \end{aligned}$$

since $F_{AF}^1(F_{AF}^0(\emptyset)) = F_{AF}^2(F_{AF}^1(F_{AF}^0(\emptyset)))$, then $GE_{AF} = \{a, c\}$. Therefore the grounded extension of AF is $\{a, c\}$.

In (Dung 1995), it was suggested a general method for generating metainterpreters in terms of logic programming for argumentation systems. This is the first approach which regards an argumentation framework as a logic program. This metainterpreter is divided in two units: Argument Generation Unit (AGU), and Argument Processing Unit (APU). The AGU is basically the representation of the attacks in an argumentation framework and the APU consists of two clauses. In order to define these clauses, let us introduce the predicate $d(x)$, where the intended meaning of $d(x)$ is: “the argument x is *defeated*” and the predicate $acc(x)$, where the intended meaning of $acc(x)$ is: “the argument x is *acceptable*”.

$$(C1) \quad acc(x) \leftarrow not \ d(x) \qquad (C2) \quad d(x) \leftarrow attack(y, x), acc(y)$$

The first one (C1) suggests that the argument x is acceptable if it is not defeated and the second one (C2) suggests that an argument is defeated if it is attacked by an acceptable argument. Formally, the Dung's metainterpreter is defined as follows:

³ The skeptical version of the preferred semantics is defined by the intersection of all the preferred extensions of an argumentation framework.

Definition 8 Given an argumentation framework $AF = \langle AR, attacks \rangle$, P_{AF} denotes the logic program defined by $P_{AF} = APU + AGU$ where $APU = \{C1, C2\}$ and $AGU = \{attacks(a, b) \leftarrow \top \mid (a, b) \in attacks\}$.

For each extension E of AF , $m(E)$ is defined as follows:

$$m(E) = AGU \cup \{acc(a) \mid a \in E\} \cup \{d(b) \mid b \text{ is attacked by some } a \in E\}$$

Based on P_{AF} , Dung was able to characterize the stable semantics and the grounded semantics.

Theorem 1 Let AF be an argumentation framework and E be an extension of AF . Then

1. E is a stable extension of AF if and only if $m(E)$ is an answer set of P_{AF} .
2. E is a grounded extension of AF if and only if $m(E) \cup \{not\ defeat(a) \mid a \in E\}$ is the well-founded model of P_{AF} .

This result is really important in argumentation semantics; in fact it has at least two main implications:

1. It defines a general method for generating metainterpreters for argumentation systems and
2. It defines a general method for studying abstract argumentation semantics' properties in terms of logic programming semantics' properties.

As we can see, the study of abstract argumentation semantics in terms of logic programming semantics has important implications.

3 An application of argumentation theory

In order to show the utility of argumentation theory in real applications, we present an application of argumentation theory in the contest of *human organ transplanting*.

Organ transplants are among the most complex medical procedures performed today. At this time, most donated organs and tissues come from patients who are pronounced brain dead as result of disease or injury but also from non-heart-beating donors, and living donors. Behind these medical triumphs, though, lies a fundamental problem.

There are far too few organs available for transplantation: at the time of writing this paper, at least ten people die daily due to the shortage of transplantable organs.

There are two issues that make transplantation management a very complex issue: (i) scarcity of donors, so it is important to try to *maximize* the number of successful transplants (ii) improve donor/recipient matching, because of the diversity and multiplicity of genetic factors involved in the response to the transplant.

In (Vázquez-Salceda *et al.* 2003), it was proposed an agent-based architecture called CARREL for carrying out the following tasks involved in managing the vast amount of data to be processed: 1) recipient selection (*e. g.* from patient waiting lists and patient records), 2) organ/tissue allocation (based on organ and tissue records), 3) ensuring adherence to legislation, 4) following approved protocols, 5) preparing delivery plans (*e. g.* using train and airline schedules).

Actually, only the transplant coordinator unit which has the potential donor is involved in the process of deciding whether an organ is viable or not. However, it is not rare that doctors disagree in deciding if an organ is viable or not. For instance, organs from a donor infected with endocarditis are usually discarded, even though in the literature we can find successful transplantation from donors infected with this disease (Caballero *et al.* 2005).

In (Cortés *et al.* 2005; Tolchinsky *et al.* 2005; Nieves *et al.* 2006a; Tolchinsky *et al.* 2006), it was introduced *argumentation theory* in the transplantation process of CARREL with the idea of maximizing the number of viable organs. The main idea, in this procedure, is that most specialists in transplantations (Transplant Coordinators) take part on the decision if an organ is viable or not for transplanting. This idea is based on the premises that organs are rarely non-viable or ideal

per se and in areas as the medical domain, the qualified professional often disagree. This means that what may be a sufficient reason for discarding an organ for some qualified professional may not be the same for others.

3.1 Discarding an organ in CARREL

The discarding process in CARREL is carried out by different agents that play different roles. To simplify the description we will only point out the agents whose participation is more relevant for the proposed process, while omitting the agents that play secondary roles. We also omit matters concerning security measures, such as protection of privacy that, although crucial for the applicability of the procedure are expendable to understand the overall process.

In UCT_x, the agency that represents a CARREL affiliated hospital; we can identify the Transplant Coordinator Agent (TCA_x) and the Transplant Unit Agent (TUA_x). Among their tasks, they are responsible of sending and receiving the arguments to and from the Transplant Coordinator (TC) and Transplant Unit (TU). The Mediator Agent (MA) that belongs to CARREL is in charge of evaluating the interchanging arguments. The OCATT⁴ Agent (OA) will play the transplant organization role in CARREL. For the sake of simplicity, we will use OA to name the agents that represent any of the transplant organizations, such as OCATT or ONT given that their role in CARREL is essentially the same.

The process will start as it currently does; the TC detects a potential donor and determines which of the transplantable organs are viable and which are not.

Remark 2 The TC, located at UCT_i, will provide TCA_i with a *justification* to why she believes an organ should be considered as viable or not, this is done for each organ. At this stage of the transplant process CARREL considers the organs of a donor as independent.

TCA_i will carry the information of an organ to CARREL; this information contains the justification produced by TC as well as the organ's and donor's characteristics, such as the organ type, the organ size, the donor's blood type and the donor's age, *etc.* Once TCA_i enters CARREL, having passed the security protocols, it enters the transplant organization room where it meets an OA (see Fig. 2), representing in this case the OCATT. OA, only on the basis of the organ characteristics will determine whether the organ meets the local policy criteria and if it does, it checks whether the organ characteristics matches any of the potential recipient's needs. If so, the organ is accepted, otherwise, OA will address TCA_i to an agent representing the following transplant organization, in this case the ONT. This new agent will play the same role but with the difference that the organ discarding policies may vary, and the potential recipients waiting list are different. If all the organizations fail to accept the organ, the organ is discarded. Otherwise OA, sends TCA_i to the evaluation room. If the organ offered by TCA_i cannot be transplanted under no circumstances, for instance if the organ has a malignant tumour, the organ is discarded at the first instance by OA.

In the evaluation room TCA_i meets MA that will send a notification to all the UCT_x that have potential recipients waiting for an organ with the same characteristics as the offered organ. Each notified TU, in UCT_j, will send a TUA_j that will provide MA with a justification to their decision indicating why they consider they should accept or refuse that organ. Their justification is built as a response to TCA_i's justification.

If both TCA_i and TUA_j agree, that is, they both consider the organ either viable or non-viable, MA accepts their decision. But if they disagree, MA evaluates TUA_j's arguments, on the basis of TCA_i's arguments, and if it accepts them, it is TUA_j's decision that prevails, otherwise, it is TCA_i's decision which prevails.

This should be happening simultaneously with all the TUs and for all the transplantable organs of the potential donor. Hence, for each TUA_j, after MA's evaluation, a given organ can be labelled as viable or non-viable depending on the arguments they have provided. In particular, an organ initially offered as non-viable, can be labelled as viable by a TUA_j. In the current discarding process, this organ would have never been offered, preventing many potential recipients from the possibility of benefiting from it. The proposed process not only enables augmenting the human organ pool, but it also has an effect on the *allocation process*.

⁴ OCCATT and ONT represent two transplant organizations whose objective is to administrate and legislate the organ transplanting process.

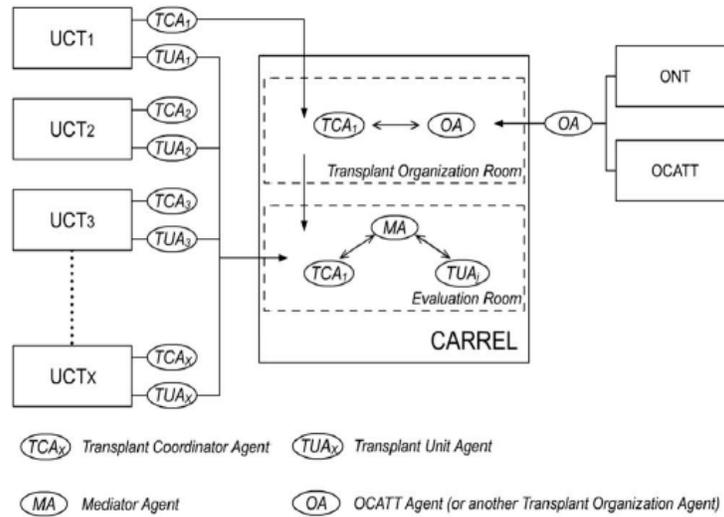


Fig. 2. The Discard Process managed by CARREL

TUAs to which an organ has been labelled as viable are committed to accept the organ and to successfully transplant it. If a committed TUA_j fails to accept or successfully transplant the organ it will have to justify its decision or action to CARREL. CARREL's policy is to promote the transplantation of as many organs as possible as long as they are safe, *i. e.* as long as the organs are viable. Thus,

Remark 4 Any refusal to transplant an organ should be justified.

Also, transplant operation must be safe, it is worth noticing that when a transplant operation failure occurs, not only the recipients health is jeopardized, but also the unsuccessfully transplanted organ will most probably be wasted, preventing another potential recipient from benefiting from it. Hence, any failure in the transplant operation must also be justified.

Remark 5 Based on the decisions and actions taken by each TUA_j, as well as on the arguments given to justify them, MA updates a model representing each TUA_j's reputation, in which TUAs with good reputation have usually fulfilled their commitments, thus, have a record of accepting the organs and successfully transplant them, while TUAs with bad reputation have a record of breaking their commitment.

Providing valid justifications when breaking commitments helps improving the agents' *reputation* (while helping to improve the understanding of the domain). It is worth mentioning that there can be several valid (and acceptable) reasons for breaking a commitment. For instance, a TUA_j that initially claimed an organ to be valid may retract from its claim because the potential recipient to whom the organ was intended had suddenly developed fever⁵. If this were to happen, this TUA_j will be committed to provide CARREL with the appropriate justification.

3.2 Building justifications

It is quite obvious that a fundament reasoning process in each agent of CARREL is the construction of a justification of any decision that it is done by an agent. In this context, the authors in (Cortés *et al.* 2005; Tolchinsky *et al.* 2005; Nieves *et al.* 2006a) suggested to use Dung's approach for building justification and for the interaction between the agents. Let us consider the following example, in order to illustrate how an agent can build a justification.

⁵ Transplant operation must not take place on patient with fever.

Let us assume that we have two transplant coordination units, one which is against the viability of the organ (UCT_D) and one which is in favour of the viability of the organ (UCT_R).

- UCT_D argues that the organ is not viable, since the organ donor had endocarditis due to *streptococcus viridans*, then the organ recipient could be infected by the same microorganism.
- In contrast, UCT_R argues that the organ is viable, because the organ presents correct function and correct structure and the infection could be prevented with a post-transplanted-treatment with penicillin, even if the organ recipient is allergic to penicillin, there is the option of a post-transplanted-treatment with teicoplanin.

Formally, we have an argumentation framework $AF := \langle AR, attacks \rangle$, where AR has the following arguments:

- a.- organ is non viable.
- b.- organ is viable.
- c.- organ has correct function and correct structure.
- d.- organ recipient could be infected with streptococcus viridans.
- e.- post-transplanted-treatment with administer penicillin.
- f.- post-transplanted-treatment with administer teicoplanin.
- g.- recipient is allergic to penicillin.

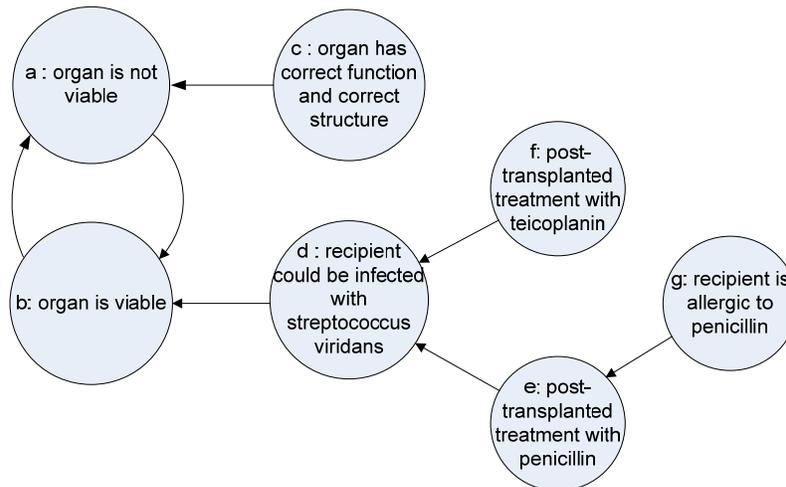


Fig. 3. A medical scenario where the decision about whether an organ from a donor with endocarditis is viable or not for being transplanted should be made. This scenario is captured by the argumentation framework

$$AF = \langle \{a, b, c, d, e, f, g\}, \{(a, b), (b, a), (c, a), (d, b), (e, d), (f, d), (g, e)\} \rangle$$

and $attacks := \{(a, b), (b, a), (c, a), (d, b), (e, d), (f, d), (g, e)\}$ (The graphic representation of AF is shown in Fig. 3). Now, based on the evidence of Figure 1, the question is: *Is the organ viable for transplanting?*

First of all, observe that each argument of the argumentation framework AF is an abstract concept that has no a defined structure. However, an argumentation framework is able to capture the relationships between the arguments. In fact, in this paper, we do not present any approach for defining a structure of an argument --- the interesting reader in defining a structure of an argument can see (Simari and Loui 1992; Prakken and Sartor 1996; Caminada and Amgoud 2007; Parsons and McBurney 2003; Nieves and Cortés 2006; Nieves *et al.* 2007a; Nieves 2008; Governatori *et al.* 2000, 2004). It is worth to comment that when one define the structure for an argument; one can define new relations of conflicts between arguments as the so called *undercut*. The interested reader can find in (Prakken and Vreeswijk 2002) a good discussion of the relationships between arguments when they have a structure.

We can see that the argumentation framework of Fig. 3 has a preferred extensions: $\{c, f, g, b\}$. Remember that any preferred extension is also a stable extension. In fact, this preferred extension also coincides with the grounded

extension of the argumentation framework of Fig. 3. Since the argument b is acceptable, *one can conclude that the organ is viable for transplanting* and the support of this conclusion are the arguments c, f , and g . Hence, one can say that the transplant coordination unit UCT_R is the winner of the disagreement.

Observe that we are not choosing a particular argumentation semantics for our example. In fact, to consider credulous (preferred and stable semantics) or skeptical reasoning (grounded semantics) in our medical domain for supporting a decision will depend on if the decision is done by

- an agent itself or
- a group of agents.

When an agent has to support a decision by itself, a credulous reasoning could warn to the agent of all the possible scenarios (possible solutions) *w.r.t.* a given problem. For the case of a group of agents a skeptical reasoning could be better since we are looking for a yes/not answer --- an organ is viable or not for transplanting.

So far, we have seen that argumentation theory is a suitable approach for modelling real intelligent systems. However, as we can see in Table 1, the decision problems in abstract argumentation semantics are hard.

In the following section, we will summarize some important results *w.r.t.* the preferred semantics and its relation with logic programming semantics. These results will suggest some practical methods for implementing abstract argumentation semantics in real systems as CARREL.

4 Declarative problem solving

In this section, we present an approach, suggested in (Nieves *et al.* 2005b; Nieves *et al.* 2005a), to infer abstract argumentation semantics. In particular, an encoding in order to infer admissible sets and preferred extensions is presented. This encoding is based on *the formal definition of an admissible set*, and the *enumerate* and *eliminate* approach. The *enumerate* and *eliminate* approach is a well known approach in Answer Set Programming (ASP) for declarative problem solving. This approach depends on how the possible solutions are generated and non-solutions are eliminated by testing (see (Baral 2003) for more details of the approach).

4.1 Admissible sets

As we know by Definition 3, a set of arguments S is admissible if and only if each argument in S is acceptable *w.r.t.* S . Hence, in order to characterize the admissible sets of an argumentation framework in a declarative way, we will specify that basic conditions that must satisfy any argument that belongs to an admissible set. Therefore, following the *enumerate* and *eliminate* approach we first need to enumerate the sets of arguments which could be admissible. In this encoding, we use the predicates $argument(a_i)$, $argument(a_j)$ and $attacked(a_i, a_j)$ to represent that the argument a_j is attacked by the argument a_i (let us denote this encoding by $\Pi(AF)$).

Declaration: We have the domain specifications.

$$argument(a_1) \leftarrow \dots argument(a_m) \leftarrow \dots attacked(a_i, a_j) \leftarrow \dots attacked(a_k, a_l) \leftarrow \dots$$

Enumeration: The enumeration rules create the possible sets which could be admissible sets. We enumerate the possibility space which specifies that each argument a_j may or may not be admissible. The rules with their intuitive meaning are as follows:

- For each argument x , either x is admissible or not.
 $admissible(x) \leftarrow not\ not_admissible(x), argument(x).$
 $not_admissible(x) \leftarrow not\ admissible(x), argument(x).$

Elimination: We use the elimination constraints to force that each admissible argument cannot be attacked by an admissible argument, and an admissible argument should be an acceptable argument.

- An admissible argument y cannot be attacked by an admissible argument x .
 $\leftarrow admissible(x), admissible(y), attacked(x, y).$
- An admissible argument x cannot be a *not_acceptable* argument.
 $\leftarrow admissible(x), not_acceptable(x), argument(x).$

- An argument x is *not_acceptable* if it is attacked by an argument y such that y is not *attacked_by_an_admissible* argument.
 $not_acceptable(x) \leftarrow attacked(y, x), not_attacked_by_pref(y), argument(x), argument(y).$
 $attacked_by_pref(y) \leftarrow argument(y), admissible(x), attacked(x, y).$

An important property of this encoding is that the sets of arguments a_i of the predicate $admissible(a_i)$, for each answer set of $\Pi(AF)$, correspond to the conflict-free sets of AR that are *admissible*. In fact, this property was formalized with the following lemma.

Lemma 1 (Nieves *et al.* 2005b; Nieves *et al.* 2005a) Let $AF = \langle AR, attacks \rangle$ be an argumentation framework, $E \subseteq AR$, $Adm := \{admissible(x) | x \in E\}$, and $Adm' := \{admissible(x) | x \in AR\}$. E is an admissible set of AF if and only if M is an answer set of $\Pi(AF)$ such that $Adm = M \cap Adm'$.

Example 2 Let us consider the argumentation framework AF which corresponds to Figure 3. Hence $AF = \langle AR, Attacks \rangle$, where $AR = \{a, b, c, d, e, f, g\}$ and $attacks := \{(a, b), (b, a), (c, a), (d, b), (e, d), (f, d), (g, e)\}$. Then, the domain specification of the program $\Pi(AF)$ is defined according to AF . $\Pi(AF)$ has twelve answer set sets; however, we only present some of them after intersecting them with the set Adm' :

$\{\}, \{admissible(ap)\}, \{admissible(cfs)\}, \{admissible(cfs), admissible(pt)\}, \dots$

As $\Pi(AF)$ has twelve answer set sets, this means that that AF has twelve admissible sets: $\{\}, \{g\}, \{c\}, \{f\}, \{c, f\}, \{b, f\}, \{c, g\}, \{f, g\}, \{v, f, g\}, \{b, c, f\}, \{c, f, g\}$ and $\{b, c, f, g\}$.

4.2 Preferred extensions

Following the case of use of the *enumerate* and *eliminate* approach we want to extend our encoding $\Pi(AF)$ in order to find the preferred extensions of an argumentation framework. The preferred extensions, by definition, are the maximal admissible sets of an argumentation framework, so we have to extend the *elimination constraints* of $\Pi(AF)$ in order to throw away the admissible sets which are not *maximal*.

There are several approaches to look for maximal sets in ASP. For instance, we can use Ordered Disjunctions Clauses introduced by (Brewka *et al.* 2002). He introduced the connective \times , called *ordered disjunction*, to allow an easy and natural representation of preferences and desires. While the disjunctive clause $a \vee b$ is satisfied equally by either a or b , to satisfy the ordered disjunctive clause $a \times b$, a will be preferred to b , *i. e.* a model containing a will have a better *satisfaction degree* than a model that contains b but does not contain a . For example, the natural language statement “*I prefer coffee than tea*” can be expressed as $coffee \times tea$.

In order to compute the preferred extensions of an argumentation framework by using the approach of *enumerate* and *eliminate*, we will present an encoding using ordered preferred disjunctions. In particular, we present an extension of this approach introduced in (Zepeda *et al.* 2005). For computing this encoding, one has to use PSmodels (Brewka *et al.* 2002).

Declaration: As in the formulation $\Pi(AF)$.

Enumeration: Also as in the formulation $\Pi(AF)$.

Elimination: This formulation has the elimination part of $\Pi(AF)$ plus the following extended preferred ordered disjunction clause (*all_pref* is a new predicate symbol):

$not_not_admissible(x) \times all_pref(x) \leftarrow argument(x).$

Observe that in this extended preferred ordered disjunction clause, a *negated negative literal* is used ($not_not_admissible(x)$). This kind of double negated literals are not usual in standard logic programming as Prolog; however, since in answer set programming a logic program can be regarded as a theory in *intuitionistic logic* (Pearce 1999; Osorio *et al.* 2004), double negated literals have a formal treatment. The intuition behind an extended ordered rule using negated negative literals is to indicate that we want to specify a preference ordering among the answer sets of a program with respect to an ordered list of atoms --- the interested reader can find in (Zepeda *et al.* 2005) the formal justification of the use of double negated literals in extended preferred ordered disjunctions. As the extended ordered disjunctions are not implemented in any answer sets solver, we use a mapping presented in

(Zepeda *et al.* 2005) in order to compute the extended ordered disjunctions using standard ordered disjunctions (PSMODELS).

$$\begin{array}{ll} aux(x) \times all_pref(x) \leftarrow argument(x). & \leftarrow not\ admissible(x), aux(x), argument(x). \\ aux(x) \leftarrow not\ tmp(x), argument(x). & tmp(x) \leftarrow not\ admissible(x), argument(x) \\ \leftarrow admissible(x), tmp(x), argument(x). & \end{array}$$

The importance of this encoding is that the sets of arguments a_i of the predicate $admissible(a_i)$, for each answer set of the about encoding, correspond to the preferred extensions of the argumentation framework AF . This property was formalized by the following theorem:

Theorem 2 (Nieves *et al.* 2005b; Nieves *et al.* 2005a) Let $AF = \langle AR, attacks \rangle$ be an argumentation framework, $E \subseteq AR$, $Adm := \{admissible(x) | x \in E\}$, and $Adm' := \{admissible(x) | x \in AR\}$. E is a preferred extension of AF if and only if M is an answer set of $\Pi(AF) \cup \{not\ not\ admissible(x) \times all_pref(x) \leftarrow argument(x)\}$ such that $Adm = M \cap Adm'$.

Example 3 Let us consider again the argumentation framework AF of Example 2. Then, the domain specification of the program $\Pi(AF)$ is defined according to AF . By computing the answer sets of $\Pi(AF) \cup \{not\ not\ admissible(x) \times all_pref(x) \leftarrow argument(x)\}$, we get the answer set: $\{admissible(g), admissible(f), admissible(c), admissible(b)\}$. This means that $\{g, f, c, b\}$ is a preferred extension of AF .

In this section we were focused on program development in ASP. The idea behind this section is to show that a proper encoding of the formal definitions of the Dung's semantics could be done by using an expressive syntax like ASP's syntax. The main advantage of using ASP's syntax is that there are efficient Answer Set Solvers (e. g. DLV, 1996; SMOODELS, 1995) and PSmodels (Brewka *et al.* 2002)) which could infer the answer set of the programs that encode the definitions of Dung's semantics.

5 Suitable codifications for arguing

To infer abstract argumentation semantics by considering strictly the definition of each argumentation semantics, as it was done in § 4, is just one possible approach for computing abstract argumentation semantics in non-monotonic logic programming (as Answer Set Programming, (Baral 2003)).

Now an interesting question is: is there a *concise* encoding which could define a direct relationship between the extensions (subset of arguments) of an argumentation framework AF and the models of a logic program P ? By concise encoding, we want to mean minimal *w. r. t* the number of atoms that appear in P . In this contest, the first result that we want to comment is the codification introduced by Dung.

As we commented in § 2, the Dung's codification is able to characterize the grounded semantics and the stable semantics (see Theorem 1) by considering the well-founded model (Gelder *et al.* 1991) of P_{AF} and the answer set models (Gelfond and Lifschitz 1991) of P_{AF} respectively. With these characterizations, he suggested an approach for defining a direct relationship between abstract argumentation semantics and logic programming semantics. The difficulty of this approach is to find a suitable codification able to define a relation, one to one, between the argumentation extensions (subset of arguments) of an argumentation framework and the models of a logic program. For instance, the program P_{AF} is able to characterize the grounded and stable semantics; however, to the best of our knowledge there does not exist a logic programming semantics able to characterize the preferred semantics by considering P_{AF} .

Now, what is a suitable codification in terms of logic programming for argumentation theory? Based on the fact that the grounded, preferred and stable semantics are the main semantics for the argumentation community (Bench-Capon and Dunne 2007; Prakken and Vreeswijk 2002; ASPIC:Project 2005), one can impose that a suitable

codification at least must be able to characterize these semantics. It is quite obvious that these conditions are valid only in Dung's approach, *i. e.* abstract argumentation semantics based on admissible sets.

It is worth mentioning that a suitable codification could be a useful tool for defining intermediate argumentation semantics between the grounded semantics and the preferred semantics. This means that it is possible to define an intermediate reasoning between the grounded semantics and the preferred semantics.

In the following section, we will present a concise mapping, which has some interesting properties in order to study argumentation semantics in terms of logic programming semantics.

6 A suitable codification

Recently, it was introduced a mapping which is able to characterize the grounded, stable and preferred semantics (Nieves *et al.* 2007ac; Nieves *et al.* 2008; Nieves and Osorio 2007). According to the authors, this mapping is inspired in the conditions which make an argument to be defeated --- this means that it is attacked by acceptable argument (see Definition 3). Basically, it captures two basic conditions which make an argument to be defeated. Like Dung's codification, the predicate $d(x)$ is used for defining this mapping. Remember that the intended meaning of $d(x)$ is: "the argument x is defeated".

Definition 9 Let $AF = \langle AR, attacks \rangle$ be an argumentation framework, then $\alpha(AF)$ is defined as follows:

$$\alpha(AF) = \bigwedge_{a \in AR} ((\bigwedge_{b: (b,a) \in attacks} d(a) \leftarrow \neg d(b)) \wedge (\bigwedge_{b: (b,a) \in attacks} d(a) \leftarrow \bigwedge_{c: (c,b) \in attacks} d(c)))$$

1. The first condition of $\alpha(AF)$ ($\bigwedge_{b: (b,a) \in attacks} d(a) \leftarrow \neg d(b)$) suggests that the argument a is defeated when any one of its adversaries is not defeated.
2. The second condition of $\alpha(AF)$ ($\bigwedge_{b: (b,a) \in attacks} d(a) \leftarrow \bigwedge_{c: (c,b) \in attacks} d(c)$) suggests that the argument a is defeated when all the arguments that defend⁶ a are defeated.

The conditions captured by $\alpha(AF)$ are standard settings in argumentation for defining the status of an argument. In fact by considering different strength of the arguments, some approaches define different status for an argument as it is done by approached based on *Defeasible Logic* (Governatori *et al.* 2000, 2004).

Observe that $\alpha(AF)$ captures conditions which make an argument to be defeated; hence, it is quite obvious that any argument which satisfies these conditions could not belong to an admissible set. Therefore these arguments also could not belong to a preferred/stable/grounded extension.

Example 4 Let $AF := \langle \{a, b, c\}, \{(a, b), (b, c)\} \rangle$ be an argumentation framework --- the graph representation of this argumentation framework is presented in Fig. 1. We can see that $\alpha(AF)$ is:

$$(d(b) \leftarrow \neg d(a)) \wedge (d(b) \leftarrow \top) \wedge (d(c) \leftarrow \neg d(b)) \wedge (d(c) \leftarrow d(a))$$

Observe that $\alpha(AF)$ has no propositional clauses *w. r. t.* argument a . This is essentially because $\alpha(AF)$ is capturing the arguments which could be defeated and the argument a will be always an acceptable argument.

It is worth to mention that in, (Besnard and Doutre 2004) is was introduced a codification closely related to $\alpha(AF)$.

Proposition 1 (propBesnard and Doutre 2004) Let $AF := \langle AR, attacks \rangle$ be an argumentation framework. Let $\beta(AF)$ be the formula:

$$\bigwedge_{a \in AR} ((a \rightarrow \bigwedge_{b: (b,a) \in attacks} \neg b) \wedge (a \rightarrow \bigwedge_{b: (b,a) \in attacks} (\bigvee_{c: (c,b) \in attacks} c)))$$

then, a set $S \subseteq AR$ is a preferred extension if and only if S is a maximal model of the formula $\beta(AF)$.

⁶ We say that c defends a if b attacks a and c attacks b .

In contrast with $\alpha(AF)$ which captures conditions which make an argument to be defeated, $\beta(AF)$ captures conditions which make an argument acceptable. In formally speaking, we can say that both codifications are dual in order to characterize the preferred semantics; because until the maximal models of $\beta(AF)$ correspond to preferred extensions AF , in (Nieves *et al.* 2008), it was shown that the minimal models of $\alpha(AF)$ correspond to the preferred extensions of AF .

Let us consider the following notation in order to understand some results related to $\alpha(AF)$ --- presented in (Nieves *et al.* 2008). Given an argumentation framework $AF := \langle AR, Attacks \rangle$ and $E \subseteq AR$, $f(E)$ and $compl(E)$ are defined as follows:

$$f(E) = \{d(a) | a \in E\} \qquad \qquad \qquad compl(E) = \{d(a) | a \in AR \setminus E\}$$

Observe that $f(E)$ essentially is embedding each argument a in the predicate $d(a)$ and $compl(E)$ essentially expresses the complement of E w. r. t. AR . As commented above, in (Nieves *et al.* 2008), it was proved that the minimal models of $\alpha(AF)$ correspond to the preferred extensions of AF .

Theorem 3 Let $AF = \langle AR, attacks \rangle$ be an argumentation framework and $S \subseteq AR$. S is a preferred extension of AF iff $compl(S)$ is a minimal model of $\alpha(AF)$.

This result has really important implications in argumentation systems. Essentially, it suggests that we can use any algorithm of minimal models for inferring the preferred extension of an argumentation framework. There are several well-known approaches for inferring minimal models from a propositional formula (Dimopoulos and Torres 1996; Ben-Eliyahu-Zohary 2005).

Another interesting results presented in (Nieves *et al.* 2008) is that one can infer preferred extensions by considering UNSAT algorithms.

Theorem 4 Let $AF = \langle AR, attacks \rangle$ be an argumentation framework and $S \subseteq AR$. S is a preferred extension of AF if and only if $compl(S)$ is a model of $\alpha(AF)$ and $\alpha(AF) \wedge SetToFormula(\neg compl(S)) \wedge \neg SetToFormula(compl(S))$ is unsatisfiable.

Observe that this result provides a method for computing preferred extensions based on Unsatisfiability (UNSAT). UNSAT is the complement of Satisfiability (SAT), a problem for which very efficient systems have been developed in AI during the last decade --- in <http://www.satcompetition.org/> the interested reader can find some of the most faster SAT solvers.

When $\alpha(AF)$ is regarded as a logic program, it is also possible to characterize the preferred semantics. In fact in (Nieves and Osorio 2007), it was shown that the pstable models of $\alpha(AF)$ also correspond to the preferred extensions of AF .

Theorem 5 Let AF be an argumentation framework and E a set of arguments. E is a preferred extension of AF if and only if $compl(E)$ is a pstable model of $\alpha(AF)$.

There is a pstable semantics solver which could be also considered for inferring the preferred semantics, (López 2006).

So far, we have seen that $\alpha(AF)$ is a suitable codification for inferring the preferred semantics; however, in (Nieves and Osorio 2007; Nieves 2008), it was also proved that when $\alpha(AF)$ is regarded as a logic program, it possible to characterize the grounded and stable semantics.

Theorem 6 Let $AF := \langle AR, attacks \rangle$ be an argumentation framework and $S \subseteq AR$.

- S is the grounded extension of AF if and only if $\exists D \subseteq AR$ such that $\langle f(D), f(S) \rangle$ is the well-founded model of $\alpha(AF)$.
- S is a stable extension of AF if and only if $compl(S)$ is a answer set of $\alpha(AF)$.

Observe that like in Theorem 1, in this theorem the grounded and stable semantics are characterized by the well-founded semantics and the answer set semantics respectively.

6.1 Preferred semantics and answer set semantics

In (Nieves *et al.* 2008; Nieves 2008), it was introduced another interesting characterization of the preferred semantics in terms of answer set models. This characterization is based on the fact that $\alpha(AF)$ is logically equivalent to the positive logic program Γ_{AF} (defined below). It is well-known that given a positive logic program P , all the minimal models of P correspond to the answer sets of P (see Gelfond and Lifschitz 1988). This property will be enough for characterizing the preferred semantics by the *answer set models* of the positive disjunctive logic program Γ_{AF} . This characterization will suggest another option for computing preferred extensions based on answer set solvers. This approach presents an easy-to-use form for inferring the preferred extensions of an argumentation framework. In this case, the kind of systems that we need for inferring the preferred extensions of an argumentation framework is any *disjunctive answer set solver* e. g. DLV, 1996.

In order to present the characterization of the preferred semantics in terms of answer sets, the following mapping is introduced

Definition 10 (Nieves *et al.* 2008) Let $AF := \langle AR, attacks \rangle$ be an argumentation framework and $a \in AR$. We define the transformation function $\Gamma(a)$ as follows:

$$\Gamma(a) := \left\{ \bigcup_{b:(b,a) \in attacks} \{d(a) \vee d(b)\} \right\} \cup \left\{ \bigcup_{b:(b,a) \in attacks} \{d(a) \leftarrow \bigwedge_{c:(c,b) \in attacks} d(c)\} \right\}$$

Now the function Γ is defined in terms of an argumentation framework.

Definition 11 (Nieves *et al.* 2008) Let $AF := \langle AR, attacks \rangle$ be an argumentation framework. We define its associated general program as follows:

$$\Gamma_{AF} := \bigcup_{a \in AR} \Gamma(a)$$

Notice that $\alpha(AF)$ is similar to Γ_{AF} . The main syntactic difference of Γ_{AF} w. r. t. $\alpha(AF)$ is the first part of Γ_{AF} which is $(\bigcup_{b:(b,a) \in attacks} (d(a) \vee d(b)))$; however this part is logically equivalent to the first part of $\alpha(AF)$ which is $(\bigcup_{b:(b,a) \in attacks} d(a) \leftarrow not\ d(b))$. In fact, the main difference is their behaviour w. r. t. answer set semantics.

The preferred semantics in terms of answer sets is defined as follows:

Theorem 7 (Nieves *et al.* 2008) Let $AF := \langle AR, attacks \rangle$ be an argumentation framework and $S \subseteq AR$. S is a preferred extension of AF if and only if $compl(S)$ is an answer set of Γ_{AF} .

One of the relevant points of this result is that we can take advantage of efficient disjunctive stable model solvers, e. g. the DLV System (DLV 1996), for inferring the preferred semantics. The DLV System is a successful stable model solver that includes deductive database optimization techniques and non-monotonic reasoning optimization techniques in order to improve its performance (Leone *et al.* 2002; Gebser *et al.* 2007). In fact, we can implement the preferred semantics inside object-oriented programs based on our characterization and the DLV JAVA Wrapper (Ricca 2003).

One of the advantages of characterizing the preferred semantics by using a logic programming semantics with *default negation*, it is that we can infer the acceptable arguments from the answer sets of Γ_{AF} in a straightforward form. In fact in (Nieves *et al.* 2008), it was defined a small variation of the mapping Γ_{AF} , in order to infer the acceptable arguments from the answer sets models of the logic program.

Definition 12 (Nieves *et al.* 2008) Let $AF := \langle AR, attacks \rangle$ be an argumentation framework. We define its associated general program as follows:

$$\Lambda_{AF} := \bigcup_{a \in AR} \{ \Gamma(a) \cup \{ a \leftarrow not\ d(a) \} \}$$

Notice that $\Gamma(a)$ and $\Lambda(a)$ are equivalent, the main difference between Γ_{AF} and Λ_{AF} is the rule $a \leftarrow not\ d(a)$ for each argument.

Proposition 2 (Nieves *et al.* 2008) Let $AF := \langle AR, attacks \rangle$ be an argumentation framework and $S \subseteq AR$. S is a preferred extension of AF if and only if there is a stable model M of Λ_{AF} such that $S = M \cap AR$.

7 Beyond admissible sets

As commented in all the paper, the three principal abstract argumentation semantics introduced by Dung are the grounded, preferred and stable semantics. However, these semantics exhibit a variety of problems which have illustrated in the literature (Prakken and Vreeswijk 2002; Baroni *et al.* 2005; Caminada 2005, 2006; Bench-Capon and Dunne 2007). Authors as Baroni *et al.* have suggested that in order to overcome Dung's abstract argumentation semantics problems, it is necessary to define flexible argumentation semantics which are not necessarily based on admissible sets (Baroni *et al.* 2005).

According to Baroni *et al.* 2005 the preferred semantics is regarded as the most satisfactory approach; however, they have also pointed out that the preferred semantics produces some questionable results in some cases concerning cyclic attack relations. For instance, let us consider the argumentation framework that appears in Fig. 4⁷. In this argumentation framework there are two arguments: a and b . We can see that the argument a is attacked by itself and the argument b is attacked by the argument a . Intuitively, we can expect that the argument b can be considered as an acceptable argument since it is attacked by argument a which is attacked by itself. However, the preferred semantics is unable to infer the argument b as an acceptable argument. In fact, none of the argumentation semantics suggested by Dung is able to infer the argument b as acceptable.



Fig. 4. Graph representation of the argumentation framework $AF := \langle \{a, b\}, \{(a, a), (a, b)\} \rangle$

Another interesting argumentation framework which has been commented on literature (Prakken and Vreeswijk 2002; Baroni *et al.* 2005) is presented in Fig. 5. The preferred semantics *w. r. t.* this argumentation framework is only able to infer the empty set. Some authors suggest that the argument e can be considered as an acceptable argument since it is attacked by argument d which is attacked by three arguments: a, b, c . Observe that the arguments a, b , and c form a cyclic of attacks.

⁷ This example is also commented in (Prakken and Vreeswijk 2002; Baroni *et al.* 2005).

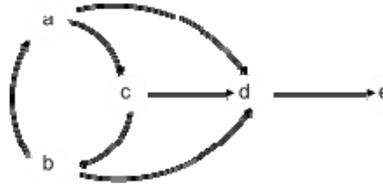


Fig. 5. Graph representation of the argumentation framework
 $AF := \langle \{a, b, c, d, e\}, \{(a, c), (c, b), (b, a), (a, d), (c, d), (b, d), (d, e)\} \rangle$

The stable argumentation semantics is also considered as a proper argumentation semantics. However, this semantics is often criticized because frequently this semantics is undefined (Caminada 2006; Bench-Capon and Dunne 2007). For instance, the stable semantics is undefined in the argumentation frameworks of Fig. 4 and Fig. 5.

The solutions to the problems of the argumentation semantics suggested by Dung are really diverse some researchers have focused on improving the stable argumentation semantics (Caminada 2006), some other researchers have focused on improving the preferred semantics (Nieves *et al.* 2006b), and still other researchers have focused on improving the concept of *admissible set* which is the basis of the argumentation semantics suggested by Dung (Jakobovits and Vermeir 1999; Baroni *et al.* 2005).

We can recognize two major branches for improving Dung's approach. On the one hand, we can take advantage of graph theory; on the other hand, we can take advantage of logic programming with negation as failure.

With respect to graph theory, the approach suggested by Baroni *et al.*, in (Baroni *et al.* 2005) is maybe the most general solution defined until now for improving Dung's approach. This approach is based on a solid concept in graph theory which is a *strong connected component* (SCC). Based on this concept, Baroni *et al.*, describe a recursive approach for generating new argumentation semantics. For instance, the argumentation semantics CF2 suggested in (Baroni *et al.* 2005) is able to infer the argument *b* as an acceptable argument of the argumentation framework of Fig. 4. Also CF2 is able to infer the extensions: $\{a, e\}$, $\{b, e\}$, $\{c, e\}$ from the argumentation framework of Fig. 5. This means that CF2 regards the argument *e* as an acceptable argument.

With respect to logic programming, in (Nieves 2008) it was presented an approach for defining new argumentation semantics which is based on regarding an argumentation framework as logic program and then apply different logic programming semantics. Some of the new argumentation semantics presented in, (Nieves 2008), are extensions of the grounded semantics. One of the approaches presented in (Nieves 2008) is based on splitting a logic program into its components (by component we understand as a subprogram). In fact, according to (Nieves 2008), this approach is inspired in the approach introduced by Baroni *et al.*, 2005. The interesting thing of this approach is that they are able to construct argumentation semantics with similar behavior to the argumentation semantics defined in terms of strong connected component defined in (Baroni *et al.* 2005). For instance, (Nieves 2008) defines the argumentation semantics MM_{Arg}^{*r} which has similar behaviour to the argumentation semantics CF2 introduced in (Baroni *et al.* 2005).

In general terms, we can see that the approach of regarding an argumentation framework as logic program does not only allow to study the well-known argumentation semantics; but also, this approach allows to explore solid solutions for the challenges that there are in argumentation theory.

8 Conclusions

Argumentation is a prominent research area that has been gaining increasing importance in the community of Computer Science. The importance of argumentation in the last years has been reflected by the number of conferences on argumentation that have been organized the last years *e. g.* The International Conferences on Computational Models of Argument (<http://www.comma-conf.org/>), the ArgMAS Workshop Series (Workshop on Argumentation and Multiagent Systems), the CMNA Workshop Series (Workshop on Computational Models of

Argument), the ArgNMR Workshop (Workshop on Argumentation and Non-monotonic Reasoning), *etc.* In fact, in the last year, we can find special editions in top journals: Artificial Intelligence (Bench-Capon and Dunne 2007) and IEEE Intelligent Systems (Rahwan and McBurney 2007). Also, there are some important projects on argumentation supported by European Commission *e. g.* Argument Service Platform with Integrated Components⁸ and ArguGRID⁹.

As young area, in argumentation there are many challenges in order to build real intelligence systems based on fundamental mechanisms of argumentation (Bench-Capon and Dunne 2007). In this paper, we concentrate our attention in some well-accepted patterns (argumentation semantics) of inference of arguments which are part of a conflict. In particular, we summarise some concise encodings of argumentation structures (argumentation frameworks) as logic programs in order to take advantage of non-monotonic technology (*e. g.* answer set solvers, UNSAT algorithms, *etc.*) for inferring some well-accepted Dung's argumentation semantics. As we saw in Table 1, the computational complexity of the decision problem of argumentation semantics as the *stable* and *preferred* semantics is hard. Hence, to consider algorithms of general purpose as the UNSAT algorithms could help to close the huge gap between argumentation theory and argumentation systems. It is worth to comment that UNSAT is the complement of Satisfiability (SAT), a problem for which very efficient systems have been developed in AI during the last decade.

The characterization of argumentation semantics in terms of logic programming semantics does not only contribute in the inference of the well-accepted argumentation semantics but also this approach contributes to study the non-monotonic reasoning properties of the argumentation semantics. For instance, in (Nieves and Osorio 2007), the preferred semantics was characterized by the *pstable semantics*, since the *pstable semantics* can be constructed by some paraconsistent logic (Osorio *et al.* 2006, 2008), one can study the non-monotonic reasoning properties of the preferred semantics in terms of these logics. Observe that this kind of results also help to understand the close relationship between two successful approaches of non-monotonic reasoning: argumentation theory and logic programming with *negation as failure*. Hence, one interesting issue in argumentation research is to explore new characterizations of argumentation semantics in terms of logic programming semantics. In fact, since in the literature of argumentation has been exhibited a variety of problems of the grounded, stable and preferred semantics (Prakken and Vreeswijk 2002; Baroni *et al.* 2005; Caminada 2005, 2006; Bench-Capon and Dunne 2007), nowadays it has increased the number of new argumentation semantics in the context of Dung's argumentation approach (Baroni and Giacomin 2007). However many of these new argumentation semantics are only motivated by particular examples; hence, the identification of the non-monotonic reasoning properties, that a particular argumentation semantics satisfies, will take relevance in order to support the well-behaviour of an argumentation semantics. In (Baroni and Giacomin 2007), it was defined a first set of basic principles in order to evaluate argumentation semantics. We believe that the set of principles described in (Baroni and Giacomin 2007) can be enriched by the identification of the non-monotonic reasoning properties that must satisfy any argumentation semantics. Of course, that this study could be explored by the characterization of argumentation semantics in terms of logic programming semantics.

In § 7, a novel strategy for constructing argumentation semantics in terms of logic programming semantics is described. This approach is able to suggest new argumentations semantics that are one hundred percent based on logic programming semantics with *negation as failure*. An interesting feature of this approach is that this approach can construct argumentation semantics with similar favour to argumentation semantics which are based on graph theory *e. g.* MM_{Arg}^* , CF2.

In general, the study of argumentation semantics in terms of logic programming semantics is promised. A good study of the relationship between argumentation semantics and logic programming semantics with negation as failure could contribute to develop of prominent non-monotonic reasoning approaches.

⁸ <http://www.argumentation.org/>

⁹ <http://www.argugrid.org/>

Acknowledgement

We are grateful to anonymous referees for their useful comments. J.C. Nieves thanks to CONACyT for his PhD Grant. J.C. Nieves and U. Cortés would like to acknowledge support from the EC funded project SHARE-it: Supported Human Autonomy for Recovery and Enhancement of cognitive and motor abilities using information technologies (FP6-IST-045088). The views expressed in this paper are not necessarily those of the SHARE-it consortium.

References

1. **Alsinet T., Chesñevar C., Godo L., Sadri S. and Simari G.** Formalizing argumentative reasoning in a possibilistic logic programming setting with fuzzy unification. *International Journal of Approximate Reasoning*, 18(3): 711-729, 2008.
2. **ASPIC:Project.** *Deliverable D2.2:Formal semantics for inference and decision-making*. Argumentation Service Platform with Integrated Components, 2005.
3. **Baral C.** *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press, Cambridge, 2003.
4. **Baroni P. and Giacomin M.** On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence.*, 171(10-15): 675-700, 2007.
5. **Baroni P., Giacomin M., and Guida G.** SCC-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence*, 168: 162-210, October 2005.
6. **Ben-Eliyahu-Zohary R.** An incremental algorithm for generating all minimal models. *Artificial Intelligence*, 169(1): 1-22, 2005.
7. **Bench-Capon T. J. M. and Dunne P. E.** Argumentation in artificial intelligence. *Artificial Intelligence*, 171 (10-15): 619-641, 2007.
8. **Besnard P. and Doutre S.** Checking the acceptability of a set of arguments. In *Tenth International Workshop on Non-Monotonic Reasoning (NMR 2004)*, pages 59-64, June 2004.
9. **Brewka G., Niemelä I., and Syrjänen T.** Implementing Ordered Disjunction Using Answer Set Solvers for Normal Programs. In *Proceedings of the 8th European Workshop Logic in Artificial Intelligence JELIA 2002*, Springer, 2002.
10. **Caballero F., López-Navidad A., Perea M., Cabrer C., Guirado L. and Solá R.** Successful liver and kidney transplantation from cadaveric donor with left-sided bacterial endocarditis. *American Journal of Transplantation*, 5: 781-787, 2005.
11. **Caminada M.** Contamination in formal argumentation systems. In *BNAIC 2005 - Proceedings of the Seventeenth Belgium-Netherlands Conference on Artificial Intelligence*, Brussels, Belgium, October 17-18, pages 59--65, 2005.
12. **Caminada M.** Semi-Stable semantics. In P.E. Dunne and T. J. Bench-Capon, editors, *Proceedings of COMMA*, volume 144, pages 121--130. IOS Press, 2006.
13. **Caminada M. and Amgoud L.** On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171: 286-310, 2007.
14. **Cayrol C., Doutre S. and Mengin J.** On Decision Problems related to the preferred semantics for argumentation frameworks. *Journal of Logic and Computation*, 13 (3): 377-403, 2003.
15. **Chesñevar C., Maguitman A. G., and Loui R. P.** Logical models of argument. *ACM Comput. Surv.*, 32(4):337-383, 2000.
16. **Cortés U., Tolchinsky P., Nieves J.C., López-Navidad A., and Caballero F.** Arguing the discard of organs for transplantation in CARREL. In *CATAI 2005*, pages 93--105, 2005.
17. **Dimopoulos Y. and Torres A.** Graph theoretical structures in logic programs and default theories. *Theor. Comput. Sci.*, 170(1-2): 209--244, 1996.
18. **DLV.** *Vienna University of Technology*. <http://www.dbai.tuwien.ac.at/proj/dlv/>, 1996.

19. **Doutre S. and Mengin J.** Preferred Extensions of Argumentation Frameworks: Computation and Query Answering. In R. Goré, A. Leitsch, and T. Nipkow, editors, *IJCAR 2001*, volume 2083 of *LNAI*, pages 272-288. Springer-Verlag, 2001.
20. **Dung P. M.** On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2): 321-358, 1995.
21. **Dung P. M., Kowalski R. A., and Toni F.** Dialectic proof procedures for assumption-based, admissible argumentation. *Artificial Intelligence*, 170(2): 114--159, 2006.
22. **Dung P. M., Mancarella P., and Toni F.** Computing ideal sceptical argumentation. *Artificial Intelligence*, 171(issues 10-15): 642--674, 2007.
23. **Dunne P.E.** On concise encodings of preferred extensions. In *9th International Workshop on Non-Monotonic Reasoning (NMR 2002)*, April 19-21, Toulouse, France, *Proceedings*, pages 392--398, 2002.
24. **Dunne P. E.** Computational properties of argument systems satisfying graph-theoretic constraints. *Artificial Intelligence*, 171(10-15): 701--729, 2007.
25. **Dunne P.E. and Bench-Capon T. J.M.** Complexity in value-based argument systems. In *JELIA*, volume 3229 of *LNCS*, pages 360--371. Springer, 2004.
26. **Egly U. and Woltran S.** Reasoning in Argumentation Frameworks Using Quantified Boolean Formulas. In Dunne P.E. and Bench-Capon T.J., editors, *Proceedings of COMMA*, volume 144, pages 133--144. IOS Press, 2006.
27. **Geber M., Liu L., Namasivayam G., Neumann A., Schaub T., and Truszczynski M.** The first answer set programming system competition. In Brewka G. Baral C. and Schlipf J., editors, *Ninth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-07)*, number 4483 in *LNAI*, pages 3--17. Springer-Verlag, 2007.
28. **Gelder A. V., Ross K. A., and Schlipf J. S.** The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3): 620--650, 1991.
29. **Gelfond M. and Lifschitz V.** The Stable Model Semantics for Logic Programming. In Kowalski R. and Bowen K., editors, *5th Conference on Logic Programming*, pages 1070--1080. MIT Press, 1988.
30. **Governatori G., Maher M. J., Antoniou G., and Billington D.** Argumentation Semantics for Defeasible Logic. *Journal of Logic and Computation*. 14(5): 675-702, 2004.
31. **Governatori G. and Maher M. J.** An Argumentation-Theoretic Characterization of Defeasible Logic. *ECAI*, pp 469-473, 2000.
32. **Gelfond M. and Lifschitz V.** Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing*, 9: 365--385, 1991.
33. **Hitchcock D. and Verheij B.** Arguing on the Toulmin Model. Volume 10 of *New Essays in Argument Analysis and Evaluation*, Springer, 2007.
34. **Jakobovits H. and Vermeir D.** Robust semantics for argumentation frameworks. *Journal of logic and computation*, 9(2): 215--261, 1999.
35. **Leone N., Pfeifer G., Faber W., Calimeri F., Dell'Armi T., Eiter T., Gottlob G., Ianni G., Ielpa G., Koch C., Perri S., and Polleres A.** The dlv system. In *JELIA*, pages 537--540, 2002.
36. **López A.** Implementing pstable In Dávila R., Osorio M., and Zepeda C., editors, *Workshop in Logic, Language and Computation*, volume 220. CEUR Workshop Proceedings, 2006.
37. **Martinez D. C., Gracia A. J., Simati G. R.** On acceptability in abstract argumentation frameworks with an extended defeat relation. In *Proc. COMMA 2006*, in *Frontiers in Artificial Intelligence and Applications*, vol 144, IOS Press, pp 273-278, 2006.
38. **Nielsen S.H. and Parsons S.** Computing Preferred Extensions for Argumentation Systems with Sets of Attacking Arguments. In P. E. Dunne and T. J. Bench-Capon, *Proceedings of COMMA*, volume 144, pages 97-108. IOS Press, 2006.
39. **Nieves J. C.** Modeling arguments and uncertain information --- A non-monotonic reasoning approach. PhD thesis, Software Department (LSI), Technical University of Catalonia, 2008.

40. **Nieves J. C. and Cortés U.** Modality argumentation programming. In *Proceedings of the conference Modeling Decisions for Artificial Intelligence (MDAI 2006)*, volume 3885 of *Lecture Notes in Computer Science*, pages 295-306. Springer, 2006.
41. **Nieves J. C. and Osorio M.** Inferring preferred extensions by pstable semantics. In *Proceedings of the Latin-American Workshop on Non-Monotonic Reasoning (LA-NMR07) Workshop*, volume 286 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.
42. **Nieves J. C., Osorio M., and Cortés U.** Modeling argumentation based semantics using non-monotonic reasoning. Research report LSI-05-59-R, Universitat Politècnica de Catalunya, Software Department, Barcelona, Spain, December 2005a.
43. **Nieves J. C., Osorio M., Zepeda C., and Cortés U.** Inferring acceptable arguments with answer set programming. In *Sixth Mexican International Conference on Computer Science (ENC 2005)*, pages 198--205. IEEE Computer Science Press, September 2005b.
44. **Nieves J. C., Osorio M., and Cortés U.** Supporting decision making in organ transplating using argumentation theory. In *LANMR 2006: 2nd Latin American Non-Monotonic Reasoning Workshop*, pages 9-14, 2006a.
45. **Nieves J.C., Osorio M., Cortés U., Olmos I., and Gonzalez J. A.** Defining new argumentation-based semantics by minimal models. *Seventh Mexican International Conference on Computer Science (ENC 2006)*, pages 210--220. IEEE Computer Science Press, September 2006b.
46. **Nieves J. C., Osorio M., and Cortés U.** Modality-based argumentation using possibilistic stable models. In Kibble R., Grasso F., and Reed C., editors, *7th Workshop on Computational Models of Natural Argument (CMNA VII)*, pages 35--41, Hyderabad, India, January 2007a.
47. **Nieves J.C., Osorio M., and Cortés U.** Inferring preferred extensions by minimal models. In Simari G. and Torroni P., editors, *Argumentation and Non-Monotonic Reasoning (LPNMR-07 Workshop)*, pages 114--124, Arizona, USA, 2007b.
48. **Nieves J.C., Osorio M., and Cortés U.** Preferred extensions as stable models. *Theory and Practice of Logic Programming*, in press, <http://arxiv.org/abs/0803.3812>, 2008.
49. **Osorio M., Arrazola J. R., and Carballido J. L.** Logical Weak Completions of Paraconsistent Logics. *Journal of Logic and Computation*, doi: 10.1093/log-com/exn015, 2008.
50. **Osorio M., Navarro J. A., and Arrazola J.** Applications of Intuitionistic Logic in Answer Set Programming. *Theory and Practice of Logic Programming* 4(3): 325-354, 2004.
51. **Osorio M., Navarro J. A., Arrazola J. R., and Borja V.** Logics with Common Weak Completions. *Journal of Logic and Computation*, 16(6):867-890, 2006.
52. **Parsons S. and McBurney P.** Argumentation-based dialogues for agent co-ordination. *Group Decision and Negotiation*, 12 (5): 415-439, 2003.
53. **Parsons S., Wooldridge M., and Amgoud L.** Properties and complexity of some formal inter-agent dialogues. *Journal of Logic and Computation*, 13 (3): 347-376, 2003.
54. **Pearce D.** Stable inference as intuitionistic validity. *Journal of Logic Programming*, 38, 79--91, 1999.
55. **Prakken H.** Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, 5 (6): 1009--1040, 2005.
56. **Prakken H. and Vreeswijk G. A. W.** Logics for defeasible argumentation. In Gabbay D. and Günthner F., editors, *Handbook of Philosophical Logic*, volume 4, pages 219--318. Kluwer Academic Publishers, Dordrecht/Boston/London, second edition, 2002.
57. **Prakken H. Sartor G.** A dialectical model of assessing conflicting arguments in legal reasoning. *Artificial Intelligence and Law*, 3:331-368, 1996
58. **Rahwan I. and McBurney P.** Argumentation technology: Introduction to the special issue. *IEEE Intelligence Systems*, 22(6): 21--23, 2007.
59. **Reed C. and Norman T.,** editors. *Argumentation Machines*, volume 9 of *New Frontiers in Argument and Computation*. Kluwer Academic Publishers, 2004.
60. **Ricca F.** The dlv java wrapper. In *Joint Conference on Declarative Programming, AGP-2003, Reggio Calabria, Italy, September 3-5*, pages 263--274, 2003.

61. **Simari G. R. and Loui R. P.** A Mathematical treatment of defeasible argumentation and its implementation. *Artificial Intelligence*, 53:125-157, 1992.
62. **S MODELS.** *Helsinki University of Technology*. <http://www.tcs.hut.fi/Software/smodels/>, 1995.
63. **Tolchinsky P., Cortés U., Nieves J. C., López-Navidad A., and Caballero F.** Using arguing agents to increase the human organ pool for transplantation. In *Proc. of the Third Workshop on Agents Applied in Health Care (IJCAI 2005)*, 2005.
64. **Tolchinsky P., Cortés U., Modgil S., Caballero F., and López-Navidad A.** Increasing Human-Organ Transplant Availability: Argumentation-Based Agent Deliberation. *IEEE Intelligent Systems: Special Issue on Intelligent Agents in Healthcare*, 21 (5): 30--37, November/December 2006.
65. **Vázquez-Salceda J., Cortés U., Padget J., López-Navidad A., and Caballero F.** Extending the CARREL System to mediate in the organ and tissue allocation processes: A first approach. *Artificial Intelligence in Medicine*, 3: 233-258, 2003.
66. **Zepeda C., Osorio M., Nieves J.C., Solnon C., and Sol D.** Applications of preferences using answer set programming. In *Proc. of the third workshop on Answer Set Programming : Advances in theory and implementation*, Bath, UK, 2005.



Juan Carlos Nieves He is a researcher in the Knowledge Engineering and Machine Learning Group in the department of Llenguatges i Sistemes Informàtics (LSI) at the Technical University of Catalonia (UPC). Until to day, He has published more than 30 research papers in the computer science area. He has participated as reviewer in several international congresses in Artificial Intelligence and Computer Science.



Mauricio Osorio He is a titular professor in the Departamento de Computación, Electrónica y Mecatrónica at Universidad de las Americas Puebla. He belongs to Sistema Nacional de Investigadores in Mexico. He is co-author of more than 100 papers in Workshops/Conferences/Journals. He has a google page at: <http://osoriomaui.googlepages.com/>



Ulises Cortés *He is a professor of the Technical University of Catalonia (UPC) since 2007. He has been working, since 1982, on several areas of Artificial Intelligence such as: knowledge acquisition for and concept formation in knowledge-based systems, as well as on machine learning and also in autonomous intelligent agents. He has been awarded with the CLUSTER chair at the École Polytechnique Fédérale de Laussane (EPFL) for 1998-1999. He has been guest editor for special issues in several international journals as: AiCommunications, Journal of Computer-aided and Civil Engineering, Computación y Sistemas, Environmental Modelling & Software, Applied Intelligence, Neural Networks, and Complex Problem-Solving Technologies, and International Journal of Approximate Reasoning. Since July 2002 until 2008 he has been member of the European Coordinating Committee for Artificial Intelligence Board (ECCAI).*