

2014-8

## Multi-Valued Argumentation Frameworks

Pierpaolo Dondio

Technological University Dublin, pierpaolo.dondio@tudublin.ie

Follow this and additional works at: <https://arrow.tudublin.ie/scschcomcon>



Part of the [Artificial Intelligence and Robotics Commons](#), and the [Theory and Algorithms Commons](#)

---

### Recommended Citation

P. Dondio. (2014). Multi-Valued Argumentation Frameworks. *Rules on the Web. From Theory to Applications*, pg. 142-156. doi:10.1007/978-3-319-09870-8\_10

This Article is brought to you for free and open access by the School of Computer Sciences at ARROW@TU Dublin. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@TU Dublin. For more information, please contact [arrow.admin@tudublin.ie](mailto:arrow.admin@tudublin.ie), [aisling.coyne@tudublin.ie](mailto:aisling.coyne@tudublin.ie).



This work is licensed under a [Creative Commons Attribution-NonCommercial-Share Alike 4.0 License](#)

# Multi-Valued Argumentation Frameworks

Pierpaolo Dondio

School of Computing, Dublin Institute of Technology,  
Kevin Street 2, Dublin 8, Ireland  
Pierpaolo.dondio@dit.ie

**Abstract.** In this paper we explore how the seminal Dung’s abstract argumentation framework can be extended to handle arguments containing gradual concepts. We allow arguments to have a degree of truth associated with them and we investigate the degree of truth to which each argument can be considered accepted, rejected and undecided by an abstract argumentation semantics. We propose a truth-compositional recursive computation, and we discuss examples using the major multi-valued logics such as Godel’s, Zadeh’s and Łukasiewicz’s logic. The findings are a contribution in the field of non-monotonic approximate reasoning and they also represent a well-grounded proposal towards the introduction of gradualism in argumentation systems.

**Keywords:** Abstract Argumentation, multi-valued Logic, Possibility Theory

## 1 Introduction

The aim of this paper is to extend the well-studied abstract argumentation framework by Dung [2] to handle arguments containing graded and vague concepts. An abstract argumentation framework is a directed graph where nodes represent arguments and arrows represent the attack relation. These frameworks were introduced to analyse defeasible arguments and study conflict resolution strategies among them. To this end, various semantics have been proposed to identify the set of acceptable arguments. In this work we deal with grounded semantics and we follow the labelling approach proposed in [6], where a semantics assigns to each argument a label in, out or undec, meaning that the argument is considered consistently acceptable, non-acceptable or undecided.

In Dung’s original work, arguments are either fully asserted or not asserted at all, and as a consequence abstract argumentation results are often too strict and coarse to support a decision making process.

In quest for an argumentation system able to handle *numbers*, few approaches have been proposed to handle various degree of strengths (such as [7]), or gradualism [1].

Recent approaches [4,5] have tried to marry abstract argumentation and probability calculus. Following a similar conceptual framework, here we investigate how to marry abstract argumentation and multi-valued logic to handle vague arguments. In our framework each argument has a degree of truth associated with it, quantifying to which de-

gree it holds. Our last statement - *arguments hold to a degree of truth* - is at least problematic. However, there are cases where the structure of arguments is defined in a way that makes it reasonable. In general, an argument can be defined as a construct used in discussions with a support and a claim that is derived from the support. An argument could be an inference rule from a premise (support) to a conclusion (claim). Premises and conclusions could be multi-valued propositions containing graded concepts or fuzzy terms that satisfy a certain state of affairs to a degree. For instance, the rule “*if the tomato is rotten, do not eat it*” can be used as an argument to avoid eating a specific tomato, it has a premise containing the fuzzy term *rotten* and therefore different tomatoes can satisfy the premise of the rule to a different degree.

Arguments containing vague or graded concepts are involved in conflicts, even if the nature of the conflict is not as well defined as in the case of Boolean propositions. As an example of conflict, let us presume that during a legal trial witness *A* said that “*the murderer was thin*” and witness *B* said that “*the murderer was tall*”. Suspect  $S_1$  is skinny and suspect  $S_2$  is about 1.9 metres tall.

Two arguments can be put forward based on the available evidence. One, based on witness *A*’s testimony, is against  $S_1$  and the other, based on *B*’s testimony, is against  $S_2$ . Each of them is satisfied to a degree. Since both are satisfied, there is an undecided situation to some degree  $x$ . However, if  $S_2$  is taller than  $S_1$  is thinner, it could be argued – to a different degree  $y$  – that there is an undefeated argument against  $S_2$  only. However, since  $S_2$  is not *completely* tall, we might argue – to another degree  $z$  probably less than  $y$  and potentially null – that there is a consistent argument against  $S_1$  only. How the degrees  $x, y, z$  can be quantified is the aim of this work.

The paper is organized as follows. The next section provides the background definitions for abstract argumentation and multi-valued logic. Sections 3 and 4 describe our computational framework with the required examples, followed by a description of related works in section 6. A conclusion summarises the paper and highlight future works.

## 2 Abstract Argumentation

### 2.1 Background Definitions

**Definition 1** An argumentation framework  $AF$  is a pair  $(Ar, R)$ , where  $Ar$  is a non-empty finite set whose elements are called arguments and  $R \subseteq Ar \times Ar$  a binary relation, called the attack relation. If  $(a, b) \in R$  we say that  $a$  attacks  $b$  in . Two arguments  $a, b$  are **rebuttals** iff  $(a, b) \in R \wedge (b, a) \in R$ .

**Definition 2** (conflict-free).  $Args$  is conflict-free iff  $\nexists a, b \in Args \mid (a, b) \in R$ .

**Definition 3** (admissible set).  $Args$  defends an argument  $a \subseteq Ar$  iff  $\forall b \in Ar$  such that  $(b, a) \in R, \exists c \in Args$  such that  $(c, b) \in R$ .

The set of arguments defended by  $Args$  is denoted  $F(Args)$ . A set  $Args$  is *admissible* if  $Args \subseteq F(Args)$  and it is complete if  $Args = F(Args)$

An abstract argumentation semantics identifies a set of arguments that can survive the conflicts encoded by the attack relation  $R$ . We follow the labelling approach of [6], where a semantics assigns to each argument a label *in*, *out* or *undec*.

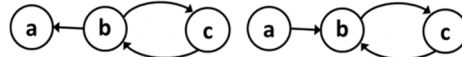
**Definition 4** (*labelling*). Let  $AF = (Ar, R)$ . A labelling is a total function  $L : Ar \rightarrow \{in, out, undec\}$ . We write  $in(L)$  for  $\{a \in Ar | L(a) = in\}$ ,  $out(L)$  for  $\{a \in Ar | L(a) = out\}$ , and  $undec(L)$  for  $\{a \in Ar | L(a) = undec\}$ .

**Definition 5** (*complete labelling*, from definition 5 in [6]). Let  $(Ar, R)$  be an argumentation framework. A complete labelling is a labelling that for every  $a \in Ar$  holds that: 1. if  $a$  is labeled *in* then all attackers of  $a$  are labeled *out*; 2. if all attackers of  $a$  are labeled *out* then  $a$  is labeled *in*; 3. if  $a$  is labeled *out* then  $a$  has an attacker labeled *in*; 4. if  $a$  has an attacker labeled *in* then  $a$  is labeled *out*.

**Theorem 1** (from [6]) Let  $L$  be a labelling of argumentation framework  $(Ar, R)$ . It holds that  $L$  is a complete labelling iff for each argument  $a \in Ar$  it holds that: 1. if  $a$  is labeled *in* then all its attackers are labeled *out*; 2. if  $a$  is labeled *out* then it has at least one attacker that is labeled *in*; 3. if  $a$  is labeled *undec* then it has at least one attacker that is labeled *undec* and it does not have an attacker that is labeled *in*.

**Theorem 2** (from theorem 6 and 7 in [7]) Given  $AF = (Ar, R)$ ,  $L$  is the grounded labelling iff  $L$  is a complete labelling where  $undec(L)$  is maximal (w.r.t. set inclusion) among all complete labellings of  $AF$ .

In figure 1 two argumentation graphs are depicted. Grounded semantics assigns the status of *undec* to all the arguments of the argumentation framework on the left, since it represents the complete labelling with the maximal set, while in the argumentation framework on the right, according to theorem 1, there is only one complete labelling (thus grounded), where argument  $a$  is *in* (no attackers),  $b$  is *out* and  $c$  is *in*. Note how  $a$  reinstates  $c$ .



**Table 1.** Fig. 1. Two Argumentation Graphs (A) and (B)

## 2.2 Subgraph Notation and Labelling of Subgraphs of an AF

As explained in section 3, when uncertainty or gradualism is added to arguments it is important to study the behaviour of a semantics over the subgraphs of the starting argumentation graph. Given an argumentation framework  $AF = (Ar, R)$  with  $|Ar| = n$ , and the graph  $\mathcal{G}$  identified by  $Ar$  and  $R$ , we consider the set  $\mathcal{H}$  of all the subgraphs of  $\mathcal{G}$ . We focus on particular sets of subgraphs, i.e. elements of  $2^{\mathcal{H}}$ .

Given  $a \in Ar$ , we define:

$$A = \{\mathcal{h} \in \mathcal{H} \mid a \text{ is a node of } \mathcal{h}\} \quad ; \quad \bar{A} = \{\mathcal{h} \in \mathcal{H} \mid a \text{ is not a node of } \mathcal{h}\}$$

$A$  and  $\bar{A}$  are respectively the set of subgraphs where argument  $a$  is present and the complementary set of subgraphs where  $a$  is not present. If  $Ar = \{a_1, \dots, a_n\}$ , a single

subgraph  $\mathcal{g}$  can be expressed by an intersection of  $n$  sets  $A_i$  or  $\bar{A}_i$  ( $i \leq n$ ) depending on whether the  $i^{\text{th}}$  argument  $a_i$  is or is not contained in  $\mathcal{g}$ . A set of subgraphs can be expressed by combining some of the sets  $A_1, \dots, A_n, \bar{A}_1, \dots, \bar{A}_n$  with the connectives  $\{\cup, \cap\}$ . We write  $AB$  to denote  $A \cap B$  and  $A + B$  for  $A \cup B$ . For instance, in figure 1 left the single subgraph with only  $b$  and  $c$  present is denoted with  $\bar{A}BC$ , while the expression  $AB$  denotes a set of two subgraphs ( $ABC$  and  $AB\bar{C}$ ) where arguments  $a$  and  $b$  are present and the status of  $c$  (not in the expression  $AB$ ) is indifferent.

Given a subgraph  $\mathcal{h} \in \mathcal{H}$ , the labelling of  $\mathcal{h}$  follows the rules of the chosen semantics. We therefore define a *subgraph labelling*  $\mathcal{L}$  as a total function over the Cartesian product of arguments in  $Ar$  and subgraphs in  $\mathcal{H}$ , therefore  $\mathcal{L}: Ar \times \mathcal{H} \rightarrow \{in, out, undec\}$ . When labelling a subgraph, we follow this choice: an argument  $a$  is automatically labelled *out* in all the subgraphs where  $a$  is not present (since it does not promote any claim) *or* when it is present but it is labelled *out* by the semantics, representing the effect on  $a$  of the other arguments. This is the only sensible choice: if an argument  $a$  is not present in a subgraph this means that  $a$  does not hold even *isolated*, since in that situation some of its premises are not satisfied. Note how, when an argument is not in the subgraph, it is a situation of perfect knowledge (we know that some of its premises are not satisfied), so it would be incorrect to assign the label *undec* or an *unknown* status to the argument. In order to be labelled *undec*, an argument has to exist and promote a claim first!

In the case of grounded semantics there is only one labelling per subgraph  $\mathcal{h}$ , that we call  $\mathcal{L}(\mathcal{h})$  (we omit  $Ar$ ). We call  $in(\mathcal{L}(\mathcal{h}))$ ,  $out(\mathcal{L}(\mathcal{h}))$ ,  $undec(\mathcal{L}(\mathcal{h}))$  the sets of arguments labelled *in*, *out*, *undec* in the labelling  $\mathcal{L}(\mathcal{h})$ . In order to study how an argument behaves across subgraphs in  $H$ , we define these sets of subgraphs:

$$\forall a \in Ar (A_{IN} = \{\mathcal{h} \in \mathcal{H}: a \in in(\mathcal{L}(\mathcal{h}))\}, A_{OUT} = \{\mathcal{h} \in \mathcal{H}: a \in out(\mathcal{L}(\mathcal{h}))\}, \\ A_U = \{\mathcal{h} \in \mathcal{H}: a \in undec(\mathcal{L}(\mathcal{h}))\})$$

i.e. the sets of subgraphs where  $a$  is labelled *in*, *out*, *undec*.

**Example 1.** In the graph of figure 1 left, there are 3 arguments and  $2^3$  subgraphs; argument  $a$  is labelled *in* in all the subgraphs where  $a$  is present and  $b$  is not present (and  $c$  becomes irrelevant), i.e.  $A_{IN} = A\bar{B}$ . It is *undec* when all the arguments are present (the single subgraph  $A_U = ABC$ ) while  $a$  is *out* when it is not present or when  $b$  is present and  $c$  is not present, i.e.  $A_{OUT} = \bar{A} + AB\bar{C}$ .

### 2.3 Computing $A_{IN}$

A brute force algorithm to find  $A_{IN}$  (or  $A_{OUT}$ ) simply computes the grounded semantics in all the subgraphs of  $Ar$  and select the subgraphs where the required label of  $a$  holds. In [18] we proposed a recursive algorithm to compute  $A_{IN}$  under grounded labelling that here we modify<sup>1</sup> to make it suitable to our problem.

---

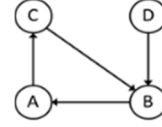
<sup>1</sup> The original algorithm in [18] generates non-overlapping sets of subgraphs containing *indifferent* arguments, as explained in section 3.

**Algorithm 1.** A is a node, L a label, P is the list of parent nodes of A.

```

FindSet(A,L,P):
if A in P:
    return empty_set //cycle found
if L = IN:
    if A terminal:
        return a //terminal condition
    else:
        add A to P
        for each child C of A
            Cset = Cset AND FindSet(C,OUT,P)
return (a AND Cset) // condition 1
if L = OUT:
    if A terminal:
        return NOT(a) //terminal condition
    else
        add A to P
        for each child C of A
            Cset = Cset OR FindSet(C,IN,P)
        return (NOT(a) OR Cset) // condition 2

```



**Fig. 2.** An argumentation graph.

Given a starting argument  $a$  and a label  $l \in \{in, out\}$ , the algorithm traverses the transpose graph (a graph with reversed arrows) from  $a$  down to its attackers, propagating the constraints of the grounded labelling. The constraints needed are listed in definition 5 and theorem 1. If argument  $a$  – attacked by  $n$  arguments  $x_n$  – is required to be labeled  $in$ , we impose the set  $A_{IN}$  to be:

$$A_{IN} = A X_{1OUT} X_{2OUT} \dots X_{nOUT} \quad (c. 1)$$

i.e. argument  $a$  can be labeled  $in$  in the subgraphs where:

1.  $a$  is present in the subgraph (i.e. the set  $A$ ) and
2. all the attacking arguments  $x_i$  are  $out$  (sets  $X_{iOUT}$ ).

If  $a$  is required to be labeled  $out$ , the set of subgraphs is:

$$A_{OUT} = \bar{A} + X_{1IN} + X_{2IN} + \dots + X_{nIN} \quad (c. 2)$$

i.e.  $a$  is labeled  $out$  in all the subgraphs where it is not present or at least one of the attackers is labeled  $in$ . Thus we recursively traverse the graph, finding the subgraphs that are compatible with the starting label of  $a$ . The sets  $X_{nOUT}$ ,  $X_{nIN}$  are found when terminal nodes are reached. When a terminal node  $x_T$  is reached the following conditions are applied:

1. if  $x_T$  is required to be  $in$  then  $X_{TIN} = X_T$
2. if node  $x_T$  is required to be  $out$  then  $X_{TOUT} = \bar{X}_T$

The way algorithm 1 treats cycles guarantees that only grounded labellings are identified. If a cycle is detected, the recursion path terminates, returning an empty set that also has the effect of discarding all the sets of subgraphs linked by a logical *AND* (in condition 1) to the cyclic path.

**Example 2.** Referring to figure 2,  $a$  is labelled *in* when:

$$A_{IN} = AB_{OUT} = A(\bar{B} + D_{IN} + C_{IN}) = A(\bar{B} + D + CA_{OUT}) = A(\bar{B} + D).$$

Note how  $CA_{OUT}$  identifies a cycle and returns the empty set.

## 2.4 Multi-valued Logic

In the setting of multi-valued logics, the convention prescribing that a proposition is either true or false is changed. A sentence is now not true or false only, but may have a truth degree taken from an ordered scale, called truth space  $S$ , such as  $[0,1]$ . Multi-valued logic can model situations affected by vagueness, where a statement is satisfied to a certain extent and the concepts discussed are graded. This is usual in natural language when words are modeled by fuzzy sets, such as *tall*, *young*, *fast*. We identify a proposition with a fuzzy set and the degree of membership of a state of affairs to this fuzzy set evaluates the degree of fit between the proposition and the state of facts it refers to. This degree of fit is called *degree of truth* of a proposition  $\phi$ . Semantically, a many-valued interpretation  $I$  maps each basic proposition  $\phi, \psi$  into  $[0,1]$  and is then extended inductively as follows:

$$\begin{aligned} I(\phi \wedge \psi) &= I(\phi) \otimes I(\psi) \quad ; \quad I(\phi \vee \psi) = I(\phi) \oplus I(\psi) \\ I(\phi \rightarrow \psi) &= I(\phi) \triangleright I(\psi) \quad ; \quad I(\bar{\phi}) = \ominus I(\psi) \end{aligned}$$

where  $\otimes, \oplus, \triangleright$  and  $\ominus$  are called triangular norms, triangular co-norms, implication functions, and negation functions, which extend the classical Boolean conjunction, disjunction, implication, and negation to the many-valued case. These functions have all to satisfy the following properties: tautology, contradiction, commutativity, associativity and monotonicity, but not all of them satisfy excluded middle ( $x \otimes \ominus x = 0$ ) or double negation ( $\ominus \ominus x = x$ ). We usually distinguish two main logics: Łukasiewicz's and Gödel's logic; the Zadeh's logic is a sublogic of Łukasiewicz's logic. Their operators are shown in table 1. For a comprehensive analysis see [16].

**Table 1.** Combination functions of various fuzzy logics

	<b>Łukasiewicz's L.</b>	<b>Gödel's logic</b>	<b>Zadeh's logic</b>
$a \otimes b$	$\max(a+b-1, 0)$	$\min(a, b)$	$\min(a, b)$
$a \oplus b$	$\min(a+b, 1)$	$\max(a, b)$	$\max(a, b)$
$a \triangleright b$	$\min(1-a+b, 1)$	$\begin{cases} 1 & \text{if } a \leq b \\ b & \text{otherwise} \end{cases}$	$\max(1-a, b)$
$\ominus a$	$1-a$	$\begin{cases} 1 & \text{if } a = 0 \\ 0 & \text{otherwise} \end{cases}$	$1-a$

### 3 Gradualism, Vagueness and Abstract Argumentation

Let us presume our argumentation framework includes  $n$  arguments and that each argument is an inference rule between propositions of a language. If these propositions are affected by uncertainty or/and vagueness, we are not sure if the claim of the argument can be used in the argumentation process. If the proposition  $\phi$  representing a claim is probabilistic, it can hold or not; if  $\phi$  is vague, it partially holds (and partially not). The consequence is that multiple scenarios of the same argumentation process are possible or should be taken into account, each scenario described by a subset of the original argumentation framework.

The case of probabilistic uncertainty has been recently analyzed in [5] and [4]. In a probabilistic argumentation framework arguments have a probability attached to them, indicating the likelihood of the argument to hold (based on the probability to which its premises are true, or are believed to be true). Since the premises are affected by probabilistic uncertainty, the premises are satisfied (and the claim follows) in a subset of situations with likelihood  $x$ , and they are not satisfied in the complementary set of situations (with likelihood  $1 - x$ ). Given an argumentation graph with  $n$  arguments, there are  $2^n$  possible situations, each of them identifying a subgraph of the original argumentation graph. Li [4] calls these situations *induced argumentation frameworks*. Each induced framework behaves as an abstract Dung-style framework and it has a probability of existing attached to it, computed using the (joint) probability distribution  $P$  defined over the arguments. Given a semantics, the probability of an argument  $a$  to be labelled *in* (or *out* or *undec*) is the sum of the probabilities of all the induced frameworks where the chosen semantics produces the required label for  $a$ . This computation is referred to in [5] as the *constellation approach*.

In a multi-valued argumentation setting, arguments have a degree of truth attached to them, indicating to which extent their claims are compatible with a state of affair. We therefore assume an underlying model of arguments as inference rules between multi-valued propositions, each proposition with a degree of truth in  $[0,1]$ . A support and/or claim of an argument might contain vague or graded terms, and they can therefore have a degree of truth when applied to a specific state of affairs. For instance, I can argue that “*if a tomato is rotten, do not eat it*”. The support and therefore the claim of the argument assumes different degrees of truth when applied to different tomatoes.

If a claim has a degree of truth  $\mu$  attached to it, this means that the current *state of affairs* satisfies the claim to a certain degree  $\mu$  but at the same time it also satisfies the negation of the claim with a degree quantified by the negation operator  $\ominus$ . These values are not referring to two distinct situations – as in the case of probabilistic uncertainty – but they represent degrees of truth attached to two co-existing situations both compatible with the same state of affairs. In a multi-valued setting, an argument always holds partially, *always* because there is no probabilistic uncertainty involved and *partially* because it can be experienced at different degrees. However, at the same time this is also true for the negation of the claim. Going back to the tomato, the tomato is rotten, but maybe *not so rotten* to avoid eating parts of it.

Given  $n$  arguments with vague claims, there are again  $2^n$  ways to which the set of arguments can partially satisfy the same state of affairs, each situation with a degree of



truth associated. In each situation we consider the degree to which some arguments satisfy the state of affairs and the others do not satisfy it. We start by defining a multi-valued argumentation framework as follows:

**Definition 6** A multi-valued argumentation framework (MVAF) is a tuple  $((Ar, R), \mu)$  where  $(Ar, R)$  is an abstract argumentation framework and  $\mu: Ar \rightarrow [0,1]$  assigns a degree of truth to each argument in  $Ar$ .

We write  $\mu_A$  as a shortcut for  $\mu(a)$ . Our aim is to find the degree to which an argument  $a$  is labelled *in* (or *out* or *undec*), called  $\mu_{A_{IN}}$  ( $\mu_{A_{OUT}}, \mu_{A_U}$ ). We stress the crucial difference between  $\mu_A$  and  $\mu_{A_{IN}}$ .  $\mu_A$  is the degree of truth to which the isolated argument  $a$  holds, *before* the argumentation process;  $\mu_{A_{IN}}$  is the resulting degree of truth of  $a$  after having accounted for the effect of the other attacking arguments

### 3.1 Computing $\mu_{A_{IN}}$

A starting idea simply translates the approach of probabilistic argumentation (the *constellation approach*) to the case of vagueness. This implies to first find all the subgraphs where  $a$  is labelled *in*, and then quantify the degree of truth of the resulting disjunction of subgraphs. Each subgraph is a conjunction of vague claims (or their negation) and its degree of truth is the degree to which this conjunction is satisfied by the state of affairs. As an example, let's consider a simple argumentation graph where argument  $a$  is attacked by  $b$ , and  $b$  is attacked by  $c$ . The constellation approach finds the following three subgraphs:  $A_{IN} = ABC + A\bar{B}C + A\bar{B}\bar{C}$ . The recursive algorithm 1 returns the following set:  $A_{IN} = AB_{OUT} = A(\bar{B} + C_{IN}) = A(\bar{B} + C) = A\bar{B} + AC$ . Note how we could also express the set  $A_{IN}$  as  $A\bar{B} + ABC$  using disjoint sets. In the probabilistic case all the above expressions are equivalent, but this is not the case for vague arguments and multi-valued logic. For instance, if  $\mu_A = 0.8, \mu_B = 0.3, \mu_C = 0.9$ , using Zadeh's *max* and *min* operators the *constellation approach* gives a value of 0.3, the recursive algorithm 0.7 and the disjoint set notation 0.8. Which computation should be preferred? Our answer is two-fold.

First, we note how the above expressions of  $A_{IN}$  are computed using classical sets operators, that are adequate if a probabilistic measure is used over arguments. However, we are not allowed to further simplify the expression of  $A_{IN}$  in case of vague arguments. The claims of the arguments are now multi-valued propositions associated to fuzzy sets, whose operators do not behave as the classical counterparts. Therefore, while the *constellation approach* implicitly assumes the classical set theory and cannot be extended to the multi-valued case, the recursive algorithm 1 could still generate a correct expression for  $A_{IN}$  if we do not simplify its output but we stop at  $A_{IN} = A(\bar{B} + C)$ . For instance, Łukasiewicz strong operators do not satisfy the distributive property and therefore the expression cannot be simplified further.

Second, it is the role of the arguments *indifferent* to the labelling of  $a$ . We set this reasonable principle: if an argument status is indifferent to the label of  $a$ , why bother considering its degree of truth? If in the probabilistic case the above question is irrelevant (since  $p(a) + p(\bar{a}) = 1$ ), it is not when dealing with vague arguments. Let's consider the *constellation approach* first. Its expression is  $A_{IN} = ABC + A\bar{B}C + A\bar{B}\bar{C}$ . In

the last two terms,  $(A\bar{B}C$  and  $A\bar{B}\bar{C})$ ,  $b$  is not in the subgraphs,  $c$  becomes disconnected from  $a$  and therefore irrelevant for the labelling of  $a$ . Therefore,  $c$ 's degree of truth should not alter the degree of truth of  $a$ . The same happens with the recursive approach using disjoint sets. In the term  $ABC$ , why should I consider  $b$ ?  $b$  is labelled *out* and therefore irrelevant for the labelling of  $a$ .

We claim that, in order to assess the degree of truth of  $A_{IN}$ , the correct expression is the one generated by algorithm 1, i.e.  $A_{IN} = A(C + \bar{B})$ , where all the arguments indifferent to the labelling of  $a$  are removed and multi-valued logic properties are not violated. Algorithm 1 directly maps the definition of complete grounded labelling as found in Caminada [6], its output is independent from the logic employed, and therefore it is correct both for the uncertain case (probabilistic or possibilistic) and the vague one.

We now show that the output of algorithm 1 does not contain *indifferent* arguments. The reasons for an argument  $b$  to be indifferent to the grounded labelling of  $a$  are the following:

1.  $b$  is disconnected from  $a$ .
2.  $b$  is in the subgraph but labeled *out* (Boella 2009).
3. If  $n$  *in*-labeled nodes are attacking an *out* node, only one attacking argument at a time is needed to label  $a$ , while the others are indifferent.

Points 1 and 3 are respected by algorithm 1. Disconnected arguments are never considered by algorithm 1 since they are simply not visited by the recursive algorithm, while the disjunction in condition 2 of algorithm 1 guarantees that only one of the attackers is considered in each term. This allows us to stress a key advantage of algorithm 1 compared to the *constellation approach*. While the constellation approach computation fragments the structure of the argumentation graphs in a collection of subgraphs, Algorithm 1 is a path-based traversal of the graph and it preserves the topology of the graph.

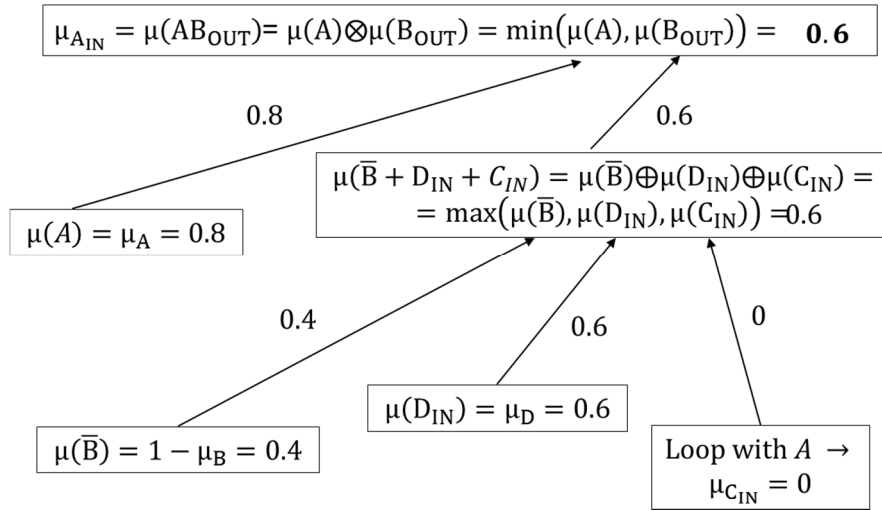
Point 2 is also verified by algorithm 1, since the last line of the algorithm (*return NOT(a) OR Cset*) is not considering argument  $a$  in its second term (since  $a$  is always labelled *out* in that case). Algorithm 1 guarantees to find a set of set subgraphs that is complete [18], i.e. its union covers all the possible subgraphs where a certain labelling of  $a$  holds.

We then exploit the fully truth-compositional nature of multi-valued logic operators. Unlike probability or possibility calculus the three multi-valued logic proposed have truth-functional operators, i.e. the degree of truth of an expression is fully determined by the degree of truth of its components. As stressed by Dubois [20], we are allowed to use truth-functional operators as long as we are dealing with gradual properties with no uncertainty involved, otherwise possibility theory has to be applied and the truth-compositional property is lost.

Therefore degrees of truth can be computed during the recursive visit of algorithm 1. Degrees of truth of arguments are found when terminal conditions are reached and the values are propagated back to the recursive step and combined with the truth-functional multi-valued logic operators. We use as conjunction, disjunction and negation the operators  $\oplus, \otimes, \ominus$  of the multi-valued logic employed, and replacing arguments with their degrees of truth when terminal conditions are met. The truth-compositional property of

multi-valued operators makes computing degrees of truth under grounded semantics having the same complexity class as a recursive tree traversal, i.e. a linear complexity proportional to the number of nodes and links, while the constellation approach is obviously of above-polynomial complexity.

**Example 3** Let us continue example 2. If  $\mu_A = \mu_C = \mu_D = 0.8, \mu_B = 0.6$  then  $\mu_{A_{IN}}$  is given by the following recursive tree:



Note how degree of truth are computed and propagated during the recursive steps exploiting the truth-compositional property of multi-valued operators applied to gradual properties with no uncertainty involved. The computation seems to consistently use both argumentation semantics and multi-valued logic.

#### 4 Attack, Reinstatement, Accrual and Rebuttals

The following examples illustrate, for all the three logics considered, the behavior of our frameworks w.r.t. fundamental situations that any argumentation framework has to handle, namely attack, reinstatement, accrual of arguments and reinstatement.

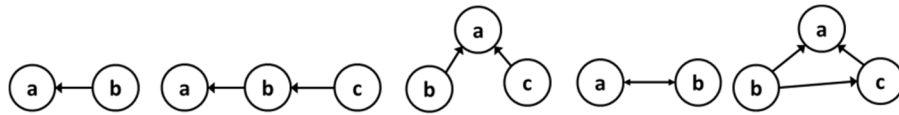


Fig. 3. Argumentation graphs for the examples 4, 5, 6, 7, 8.

**Example 4 Attack.** If argument  $a$  is attacked by  $b$ , how is the degree of  $a$  modified? It is  $A_{IN} = A\bar{B}$ . Using Zadeh's operators, it is  $\mu_{A_{IN}} = \min(\mu_A, 1 - \mu_B)$ . In general with Zadeh's operators  $\mu_{A_{IN}} < \mu_A$  (degree of truth is diminished), but it remains the same

when  $\mu_A < 1 - \mu_B$ . Therefore, the degree of truth of  $a$  could remain unchanged and the attack from  $b$  neglected if  $\mu_A + \mu_B < 1$ . This imposes a minimum degree of truth on the attacker to activate the attack. Note how this finding seems to justify the notion of a threshold for *attack activation* present in [1]. Using Łukasiewicz's logic it is:

$$\mu_{A_{IN}} = \min(\mu_A + 1 - \mu_B - 1, 0) = \min(\mu_A - \mu_B, 0) = \begin{cases} \mu_A - \mu_B & \text{if } \mu_A > \mu_B \\ 0 & \text{if } \mu_A \leq \mu_B \end{cases}$$

Therefore  $a$  is always diminished, and totally defeated if the degree of the attacker is greater than  $\mu_A$ . Interestingly, this is the exact behaviour proposed by Pollock [7], whose proposal was not grounded in any multi-valued logic system.

Note how, using Zadeh's *min* operator, an argument can be totally defeated only if  $\mu_B = 1$ , while using Łukasiewicz's logic it is totally defeated every time  $\mu_A \leq \mu_B$ .

Finally, Godel's logic negation operator always assigns a null degree of truth to  $\mu_{\ominus A}$  if  $\mu_A > 0$ . In practical terms, this implies removing the negated terms from the output of algorithm 1. This means that, using grounded semantic only one out of the three quantities  $\mu_{A_{IN}}, \mu_{A_{OUT}}, \mu_{A_U}$  has a not null value. In the case of  $b$  attacking  $a$ , it is obviously  $\mu_{A_{IN}} = 0$ .

Regarding  $\mu_{A_{OUT}}$ , it is  $A_{OUT} = \bar{A} + B$ . For Godel's logic the resulting degree is the degree of the attacker  $B$ , for Zadeh's logic  $\mu_{A_{OUT}}$  remains equal to  $\mu_A$  iff  $1 - \mu_A < \mu_B$  and under Łukasiewicz's logic  $\mu_{A_{OUT}} = 1$  ( $a$  totally defeated) when  $\mu_B \geq \mu_A$ .

**Example 5. Reinstatement Chain.** A chain of 3 arguments helps to reason about reinstatement. It is  $A_{IN} = A(\bar{B} + C)$ .

Under Godel's logic, only  $AC$  has a not null degree of truth and  $\mu_{A_{IN}} = \min(\mu_A, \mu_C)$ . Thus the argument is fully reinstated if  $\mu_C > \mu_A$  or it is reinstated to the degree equal to its defender  $c$ .

Using Zadeh's logic,  $\mu_{A_{IN}}$  is given by the expression  $\min(\mu_A, \max(1 - \mu_B, \mu_C))$ . We note that, if  $1 - \mu_B > \mu_C$ , nothing changes from example 4 and no reinstatement happens, while, when  $1 - \mu_B < \mu_C$ ,  $\mu_{A_{IN}}$  could be increased w.r.t. example 4. Both Zadeh's and Godel's logic fully reinstates  $a$  if  $\mu_C > \mu_A$ . Arguably, when  $\mu_C > \mu_A$  the two logic systems neglect the degree of truth of the attacker  $b$ .

Using Łukasiewicz's logic  $a$  is fully reinstated if  $1 - \mu_B + \mu_C > 1$ , i.e.  $\mu_C > \mu_B$ , which seems a reasonable result and again it is the same behaviour as Pollock [7].

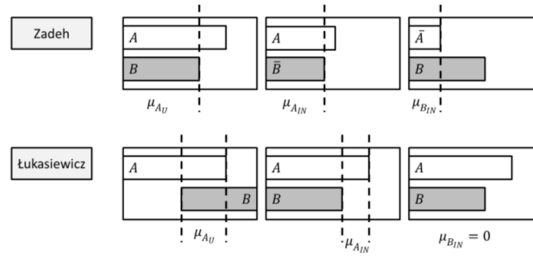
The reinstatement example provides evidence in favour of our recursive algorithm and our choices of neglecting indifferent arguments and respecting the multi-valued logic properties when simplifying the expression of  $A_{IN}$ . In fact, if we had further simplified the expression of  $A_{IN}$  into  $A_{IN} = A\bar{B} + AC$ , using Łukasiewicz's logic, it could have been that  $\mu_{A_{IN}}$  resulted more than  $\mu_A$ ! If  $\mu_A = 0.5, \mu_B = 0.1, \mu_C = 0.9$ , it is  $\mu_{A_{IN}} = \min(\max(0.5 + 0.9 - 1, 0) + \max(0.5 + 0.9 - 1, 0), 1) = 0.8$ ! We wonder if the reason why  $\mu_{A_{IN}} > \mu_A$  is because we neglected the *out*-labelled argument  $b$  in the expression  $A_{IN} = A\bar{B} + AC$ , and the right expression should be  $A_{IN} = A\bar{B} + ABC$  or the *constellation approach* expression  $A_{IN} = A\bar{B}C + A\bar{B}\bar{C} + ABC$ . Both these two expressions guarantee that  $\mu_{A_{IN}} \leq \mu_A$ , but their behaviour is still counter-intuitive due to

the fact that longer conjunctive expressions are harder to satisfy and the resulting degree of truth decreases rapidly<sup>2</sup>. For instance, if  $\mu_A = 0.5, \mu_B = 0.5, \mu_C = 1$  we have  $\mu_{A_{IN}} = 0$  (even if  $a$  is defended by an argument with the maximum degree of truth, there is no reinstatement).

**Example 6. Accrual of attacks.** The example clarifies the accrual of attacks. It is  $A_{IN} = A\bar{B}\bar{C}$  and  $A_{OUT} = \bar{A} + C + B$ . Considering  $A_{OUT}$ , both Godel's and Zadeh's operators do not accrue arguments, since it is the *max* of the two arguments that is considered, as in Pollock [7]. Arguments accrue with Łukasiewicz's logic, since its disjunction operator does.

**Example 7, Rebuttal.** In case of two rebuttal arguments, grounded semantics gives  $A_U = B_U = AB, A_{IN} = A\bar{B}, B_{IN} = B\bar{A}$ . Figure 4 shows the behaviour of the three multi-valued logics discussed. Godel and Zadeh always assign a not null value to the *undec* situation equal to  $\mu_{A_U} = \mu_{B_U} = \min(\mu_A, \mu_B)$ , while with Łukasiewicz's operators it is  $\mu_{A_U} = \max(\mu_A + \mu_B - 1, 0)$ , and therefore  $\mu_{A_U} > 0$  only when  $\mu_A + \mu_B > 1$ . Intuitively, using Łukasiewicz, two conflicting arguments can coexist if their degrees of truth are small enough to avoid overlapping.

Regarding  $\mu_{A_{IN}}$  and  $\mu_{B_{IN}}$ , Godel's system assigns a null degree of truth to both; while Zadeh's logic always assigns a not null degree, that has an upper bound in the degree to which the other conflicting argument is negated. Łukasiewicz's logic assigns a not null degree equal to  $|\mu_A - \mu_B|$  to the argument with the highest degree, and a null degree to the other. Each of this behaviour seems to fit some but not all the situations where gradual arguments conflict and the author seeks to systematically investigate this issue in the next future work.



**Fig. 4.** Rebuttals with different multi-valued logic

**Example 8.** Multi-valued operators do not always verify the excluded middle principle. This could lead to controversial situations where multi-valued argumentation strongly differs from the classical logic case. Let us consider the argumentation graph in figure 3 (last on the left). If we are using Zadeh's logic, the excluded middle principle is not verified and an argument can be at the same time present and not present in the argumentation process. It is  $\mu_{A_{IN}} = \mu(A \otimes B_{OUT} \otimes C_{OUT}) = \mu(A \otimes \bar{B} \otimes (\bar{C} \oplus B)) =$

<sup>2</sup> A similar remark was done by Pollock [7] against the use of the product rule of probability in defeasible reasoning.

$\mu(A \otimes (\bar{B} \otimes \bar{C} + \bar{B} \otimes B))$  where we applied the distributive property (allowed with Zadeh's logic) to show the presence of the not-null term  $\bar{B} \otimes B$ .

## 5 Related Works

Conceptually, our framework is closer to the work done in the context of probabilistic argumentation frameworks. The idea of merging probabilities and abstract argumentation was first presented by Dung [2], and a more detailed formalization was provided by Li [4], along with the works by Hunter [5] and Thimm [12]. [4] introduces the notion of constellation approach. [12] and [5] in his epistemic approach, start from a complementary angle. Both authors assume that there is already an uncertainty measure – potentially not probabilistic – defined on the admissibility set of each argument and they study which properties this uncertainty measure should satisfy in order to be rational. Regarding works that explicitly define fuzzy argumentation systems, we should mention the framework by Janssen [13] where fuzzy labels may be interpreted as fuzzy membership to an extension. However, [13]'s approach differs significantly from ours by the fact that the attack relation that defines the framework is taken to be fuzzy and the conflict-free and admissibility definitions are changed accordingly. In [14] a certitude factor is added to the labels *in*, *out* and *undec* as we do. The work proposes an *equational* approach to abstract argumentation, where arguments degrees have to satisfy a set of properties modelled as equations, properties that might not have any link to a fuzzy logic system. On the contrary, our computation of degrees of truth is a more consistent approach exploiting both argumentation semantics and multi-valued logics.

Regarding other works investigating gradualism in argumentation, we first mention Pollock's work on degrees of justification [7]. Pollock considers the strengths of arguments as cardinal quantities that can be subtracted. The accrual of arguments is denied and it is the argument with the maximum strength that defines the attack. It is interesting to notice how Pollock's computation is not grounded in any logic systems, but his attack function behaves like our framework using Łukasiewicz's logic, while his accrual behaves like Zadeh's and Godel's logics. The vs-defence model, by Cayrol [1], is an extension of abstract argumentation where attacks have a strength associated with them. Argument admissibility status is the result of the comparisons of attack strengths, in a way similar to our frameworks with Łukasiewicz's logic (example 1). However, there is no description about the nature and the computation of such strength. We also mention [10] that first extended Dung's framework introducing different levels of attacks. [9] proposed weighted argument systems, where attacks can have weights, and such weights might have different interpretations: an agent-based priority voting, or a measure of how many premises of the attacked argument are compromised.

## 6 Conclusions

In this paper we explored how Dung's abstract argumentation framework can be extended to handle arguments affected by vagueness. We studied some basic properties and provided examples using Godel's, Łukasiewicz's and Zadeh's multi-valued logic.

The findings are a contribution in the field of approximate reasoning and they also represent a well-grounded proposal towards the introduction of gradualism in argumentation systems. We believe to have provided a novel synthesis between argumentation semantics and gradualism, providing the theoretical foundation of a framework for reasoning under uncertainty that has both the soundness of argumentation semantics w.r.t. the identification of a consistent set of arguments, and the ability to handle gradual and vague properties proper of multi-valued logics.

The present work represents the first theoretical foundation of our framework and it opens numerous opportunities and open issues for future studies.

First, we aim to extend our frameworks to other semantics, starting from *complete* semantics such as *stable* and *preferred*.

Second, this paper presents a limited investigation and discussion on the meaning of gradual arguments and it focuses on theoretical aspects of the frameworks. What does the notion of attack with gradual arguments really mean?

A comprehensive answer requires a more structured definition of arguments and types of attacks. Further studies have to be done in investigating the various multi-value logics proposed here. In particular, the meaning of the degrees of truth computed by each multi-valued logic and which kind of vagueness each logic system is more suitable to model. It seems to the author that none of the systems studied here could reasonably handle all the situations involving vague arguments, but rather each of them captures specific situations.

Finally, work has to be done in investigating how to handle situations in which probabilistic and vague arguments coexist in the same argumentative process.

## References

1. C. Cayrol, C. D. Lagasque-Schiex. "Acceptability semantics accounting for strength of attacks in argumentation," 19th ECAI, pages 995-996. Lisbon, Portugal, 2010
2. P. Dung, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games," *Artificial Intelligence*, vol. 77, pp. 321-357, 1995
3. P. Dung, P. Thang. "Towards (Probabilistic) Argumentation for Jury-based Dispute Resolution," *COMMA 2010*. IOS Press, 171-182
4. Hengfei L., N. Oren, T. J. Norman. *Probabilistic Argumentation Frameworks*. 1st TAFE, JICAI 2011, Barcelona, Spain
5. Hunter, A. "A probabilistic approach to modelling uncertain logical arguments." *International Journal of Approximate Reasoning* (2012).
6. Caminada, M.W.A., and D.M. Gabbay. "A logical account of formal argumentation." *Studia Logica* 93.2-3 (2009): 109-145.
7. Pollock, J., "Defeasible reasoning with variable degrees of justification," 2001 *Artificial Intelligence*, Vol 133, Pg. 233-282.
8. Gabbay, M. Equational approach to argumentation networks. *Argument & Computation* 3.2-3 (2012):87-142.
9. Dunne, P.E., A. Hunter, P. McBurney, S. Parsons, "Inconsistency tolerance in weighted argument systems," in *Proc of AAMAS*, 2009.

10. Martinez, D.C., A. J. Garcia, "An abstract argumentation framework with varied-strength attacks," in Proc of KR, 2008, pp. 135–143.
11. Vreeswijk, G. "Abstract argumentation systems," *Artificial Intelligence* 90, 225–279, 1997
12. Thimm, M. "A Probabilistic Semantics for abstract Argumentation," *ECAI*. 2012.
13. Janssen, J., "Fuzzy argumentation frameworks." *Information Processing and Management of Uncertainty in Knowledge-based Systems*. 2008.
14. Gratie, C., A. M. Florea. "Fuzzy labelling for argumentation frameworks." *Argumentation in Multi-Agent Systems*. Springer 2012. 1-8.
15. M. Aurisicchio, P. Baroni, G. Bertanza, F. Toni. An Argumentation-Based Approach for Automatic Evaluation of Design Debates
16. Lukasiewicz, T., Straccia U. Managing uncertainty and vagueness in description logics for the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web* 6.4 (2008): 291-308.
17. Boella, G., Souhila K., and Van Der Torre L. "Dynamics in argumentation with single extensions: attack refinement and the grounded extension." *Proceedings of The 8th AAMAS Conference*, 2009.
18. Dondio, P. (2013). Computing the Grounded Semantics in all the Subgraphs of an Argumentation Framework. In *Computational Logic in Multi-Agent Systems* (pp. 119-137). Springer Berlin
19. Prade H., D. Dubois. "What are fuzzy rules and how to use them." *Fuzzy Sets Syst* 84 (1996): 169-185.
20. Dubois, Prade, Smeths. Gradual properties vs. uncertainty : Fuzzy logic vs. possibilistic logic, Technical report, 2000, retrieved from [http://iridia.ulb.ac.be/~psmets/Gradual\\_vs\\_Uncert.pdf](http://iridia.ulb.ac.be/~psmets/Gradual_vs_Uncert.pdf)