# Extending Modular Semantics for Bipolar Weighted Argumentation

Nico Potyka
Institute of Cognitive Science, University of Osnabrück
Osnabrueck, Germany
npotyka@uos.de

## ABSTRACT

Weighted argumentation offers a tool for decision support and social media analysis. Arguments are evaluated by an iterative procedure that takes initial weights and attack and support relations into account. Mossakowski and Neuhaus recently unified different approaches and proved first convergence results in cyclic graphs. We build up on this work, simplify and generalize convergence results and add runtime guarantees. As it turns out, there is a tradeoff between convergence guarantees and the ability to move strength values away from the initial weights. We demonstrate that continuizing semantics can avoid divergence without this tradeoff. Semantically, we extend the framework with a Duality property that assures a symmetric impact of attack and support. We also present a Java implementation of modular semantics and explain the practical usefulness of the theoretical ideas.

**ACM Reference Format:**
Nico Potyka. 2019. Extending Modular Semantics for Bipolar Weighted Argumentation . In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 9 pages.

## 1 INTRODUCTION

*Abstract argumentation* [19] allows modeling arguments and their relationships in order to decide which arguments can be accepted. Weighted bipolar argumentation frameworks start with an initial weight of arguments and adapt this weight based on the strength of their attackers and supporters [5, 11, 26, 30, 34]. Applications include decision support [10, 34], social media analysis [1, 24] and information retrieval [38]. Initial weights can be defined manually based on the reputation of arguments' sources or computed automatically based on statistics like the success rate of a source in decision support or the number of likes or retweets of an argument in social media analysis. Attack and support relations can be extracted automatically by sentiment analysis tools [1].

Mossakowski and Neuhaus recently unified different approaches by decomposing their semantics into an aggregation function that aggregates the strength of attackers and supporters and an influence function that adapts the initial weight based on the aggregate [26]. Different combinations of aggregation and influence functions yield different semantics from the literature and semantical axioms [3–5] can be related to elementary properties of these functions. They also proved first convergence results in cyclic graphs. Note that convergence is essential to obtain final strength values here. [26] gave convergence results for sum- and max-based aggregation functions and influence functions with bounded derivatives.

We will show that all convergence results from [26] are special cases of the Contraction Principle [37], generalize the results in a uniform way and add runtime guarantees. However, we also show that convergence guarantees derived from the contraction principle are bought at the expense of open-mindedness. That is, as semantics' convergence guarantees get stronger, their ability to change the initial weights gets weaker. We also give some new divergence examples based on a family of graphs from [26]. In order to avoid the tradeoff between convergence guarantees and open-mindedness, we can continuize semantics as proposed in [30]. We demonstrate that the observed divergence problems can be solved by continuization and, thus, give some additional empirical evidence for the robustness of continuous models. Subsequently, we integrate the recently introduced Duality property [30] into the framework by Mossakowski and Neuhaus by relating it to elementary properties of aggregation and influence functions. Finally, we present an implementation of Modular semantics in the Java library Attractor[1] [31] and illustrate the practical usefulness of modular semantics. All proofs can be found in the technical report [32].

## 2 BAGS AND MODULAR SEMANTICS

We consider *weighted bipolar argumentation graphs (BAGs)* as considered in [5] and [26].

*Definition 2.1 (BAG).* A *BAG* is a tuple $\mathbf{A} = (\mathcal{A}, \mathrm{w}, \mathcal{R}, \mathcal{S})$, where $\mathcal{A}$ is an $n$-dimensional vector of arguments, $\mathrm{w} \in [0,1]^n$ is a weight vector that associates an initial weight $w_i$ with every argument $\mathcal{A}_i$ and $\mathcal{R}$ and $\mathcal{S}$ are binary relations on $\mathcal{A}$ called *attack* and *support.*

The *parent vector* $g_i \in \{-1, 0, 1\}^n$ of argument $\mathcal{A}_i$ is the vector with entries $g_{i,j} = -1$ (1) iff $(\mathcal{A}_j, \mathcal{A}_i) \in \mathcal{R}$ ($(\mathcal{A}_j, \mathcal{A}_i) \in \mathcal{S}$). We visualize BAGs by means of directed graphs as in Figure 1. Nodes show the arguments with their initial weights, solid edges denote attacks and dashed edges denote supports. We let $indegree(\mathcal{A}_i) = \sum_{j=1}^n |g_{i,j}|$ be the number of attackers and supporters of $\mathcal{A}_i$.

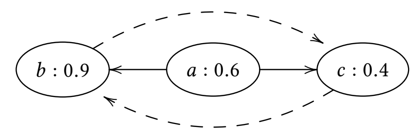


**Figure 1: Graph for Example 2.2.**

*Example 2.2.* Figure 1 shows the directed graph for the BAG $\big((a, b, c), (0.6, 0.9, 0.4), \{(a, b), (a, c)\}, \{(b, c), (c, b)\}\big)$. The parent vector of $b$ is $g_2 = (-1, 0, 1)$ and shows that $b$ is attacked by $a$ and supported by $c$. Hence, $indegree(b) = 2$.

[1] https://sourceforge.net/projects/attractorproject

*Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.* 

Given a BAG $\mathbf{A}$, we want to assign a strength value to every argument. This can be accomplished by means of different *acceptability semantics* [5]. These semantics are usually based on an iterative update procedure that may or may not converge. Therefore, we follow [26] and regard acceptability semantics as partial functions.

*Definition 2.3 (Acceptability Semantics).* An acceptability semantics is a partial function $\mathrm{Deg}_S$ that maps a BAG $\mathbf{A} = (\mathcal{A}, \mathrm{w}, \mathcal{R}, \mathcal{S})$ with $n$ arguments to an $n$-dimensional vector $\mathrm{Deg}_S(\mathbf{A}) \in [0,1]^n$ or to $\perp$ (undefined). If $\mathrm{Deg}_S(\mathbf{A}) \neq \perp$, we call the $i$-th component $\mathrm{Deg}_S(\mathbf{A})_i$ the *final strength* or *acceptability degree* of $\mathcal{A}_i$.

A *modular acceptability semantics* as introduced in [26] is an acceptability semantics that works by first aggregating the strength of attackers and supporters and then adapting the initial weight based on the aggregated value. This is accomplished by aggregation and influence functions, which satisfy some additional properties that guarantee that axioms from [5] are satisfied. Even though all axioms are interesting semantically, we will restrict to a subset here in order to keep the presentation simple and more general.

The aggregation and influence functions in [26] were supposed to be continuous. We make a stronger assumption here and assume that they are *Lipschitz-continuous.* Intuitively, this means that the growth of these functions is bounded by a constant. Lipschitz-continuity is also implied by the convergence conditions (bounded derivatives) in [26], so we do not restrict the generality of our convergence investigation. Formally, a function $f : X \to Y$ is called Lipschitz-continuous with Lipschitz constant $\lambda$ iff $\|f(x) - f(y)\|_Y \leq \lambda \|x - y\|_X$. The sets $X$ and $Y$ will contain real numbers, vectors or matrices here. We consider the maximum norm for matrices defined by $\|A\| = \max\{\sum_{j=1}^{m} |a_{i,j}| \mid 1 \leq i \leq n\}$ for an $m \times n$-matrix $A = (a_{i,j})$. That is, $\|A\|$ is the largest absolute row sum in $A$. For the special case that $x \in \mathbb{R}^n$ is a vector (an $n \times 1$-matrix), $\|x\|$ is the largest absolute value in $x$. Notice that using the maximum norm does not mean any loss of generality because all norms are equivalent in $\mathbb{R}^n$ [37] (the difference between two norms can be bounded by a constant factor).

The aggregation function requires information about the attackers and supporters, the influence function requires information about the initial weight. We regard this information as parameters of the function. We also have to express that the aggregation function depends only on the parents. As discussed in [26], this demand corresponds to the directionality axiom from [5]. In order to phrase directionality, we define an equivalence relation for every parent vector $v \in \{-1, 0, 1\}^n$. Two (strength) vectors $s_1, s_2$ are called equivalent with respect to a parent vector $v$, written as $s_1 \equiv_v s_2$ iff $s_{1,i} = s_{2,i}$ whenever $v_i \neq 0$. That is, only the strength values of parents matter, all other strength values are ignored.

In the following, for a function $f$, we let $f^k$ denote the function that is obtained by applying $f$ $k$ times, that is, $f^1 = f$ and $f^{k+1} = f^k \circ f$. Applying our update function repeatedly to the initial weights yields a sequence of strength vectors. The final strength values are defined as the limit of this sequence if it exists. Thus, convergence guarantees of update functions correspond to completeness guarantees of semantics. As usual, we say that an n-dimensional sequence $(s_n)_{n \in \mathbb{N}}$, $s_n \in \mathbb{R}^n$, converges to $s$, denoted as $\lim_{n\to\infty} s_n = s$, iff the real sequence $(\|s_n - s\|)_{n\in\mathbb{N}}$ converges to 0. That is, for every $\epsilon > 0$, there is a $N \in \mathbb{N}$ such that $\|s_m - s\| < \epsilon$ for all $m > N$. Intuitively, this means that the $i$-th component of $(s_n)$ converges to the $i$-th component of $s$.

We are now ready to define basic modular semantics.

*Definition 2.4 (Basic Modular Semantics).* A semantics $\mathrm{Deg}_S$ is called a basic modular semantics if there exists

(1) an aggregation function $\alpha_v : [0,1]^n \to \mathbb{R}$ such that for all *parent parameters* $v \in \{-1, 0, 1\}^n$ and $s, s_1, s_2 \in [0,1]^n$
   - $\alpha_v(s_1) = \alpha_v(s_2)$ whenever $s_1 \equiv_v s_2$, (Directionality)
   - $\alpha_v$ is Lipschitz-continuous, (Lipschitz-$\alpha$)
   - $\alpha_v(s) = 0$ whenever $v = 0$, (Stability-$\alpha$)

(2) an influence function $\iota_w : \mathbb{R} \to [0,1]$ such that for all *weight parameters* $w \in \mathbb{R}$
   - $\iota_w$ is Lipschitz-continuous, (Lipschitz-$\iota$)
   - $\iota_w(0) = w$ (Stability-$\iota$)

and for all BAGs $\mathbf{A} = (\mathcal{A}, \mathrm{w}, \mathcal{R}, \mathcal{S})$, we have

$$\mathrm{Deg}_S(\mathbf{A}) = \lim_{k\to\infty} f_S^k(\mathrm{w}).$$

where the $i$-th component of $f_S : [0,1]^n \to [0,1]^n$ is defined by $\iota_{\mathrm{w}_i} \circ \alpha_{g_i}$ for $i = 1, \ldots, n$. $f_S$ is called the *update function* of $\mathrm{Deg}_S$.

In practice, for the $i$-th argument $\mathcal{A}_i$, its parent vector $g_i$ serves as the parent parameter of $\alpha_v$ and its initial weight $\mathrm{w}_i$ serves as the weight parameter for $\iota_w$. Stability-$\alpha$ and Stability-$\iota$ assure that the final strength of an argument without parents will just be its initial weight. This corresponds to the stability axiom from [5].

Intuitively, modular semantics compute strength values iteratively. They start with the initial strength vector $s^{(0)} = \mathrm{w}$. Then, in the $k$-th step, the strength of argument $i$ is computed by first applying the aggregation function to $s^{(k-1)}$ and then applying the influence function to $\alpha_{g_i}(s^{(k-1)})$. That is, $s_i^{(k)} = \iota_{\mathrm{w}_i}(\alpha_{g_i}(s^{(k-1)}))$ for $k > 0$.

Table 1 shows some examples of different aggregation and influence functions that can be found in the literature.

Proposition 2.5. *The functions in Table 1 are aggregation and influence functions as defined in Definition 2.4. In particular, they are Lipschitz-continuous with the provided Lipschitz constants.*

All aggregation functions that we consider here work by computing an aggregated attack and support value independently and subtracting these values. The sum-aggregation function has been used for the Euler-based semantics in [5] and for the quadratic energy model in [30]. It aggregates strength values by adding them. The product-aggregation function is the aggregation function of the DF-QuAD algorithm [34]. Intuitively, the aggregate for attack and support is initially 1 and the aggregates are decreased by multiplying with $(1 - s)$ for an attacker or supporter with strength $s$. The top-aggregation function has been used for the top-based semantics in [4] for support-only graphs and has been generalized to bipolar graphs in [26]. It considers only the strongest attacker and supporter.

We consider three influence functions. The linear($\kappa$) influence function has a parameter $\kappa$ that we call its *conservativeness* for reasons that will become clear later. The function linear(1) can be seen as the influence function of the DF-QuAD algorithm in [34]. It moves the strength to 0 or 1 directly proportional to the aggregated strength values. This yields easily interpretable results,

| Aggregation Functions | | | |
|---|---|---|---|
| Sum | $\alpha_v^\Sigma : [0,1]^n \to \mathbb{R}$ | $\alpha_v^\Sigma(s) = \sum_{i=1}^n v_i \cdot s_i$ | $\lambda_v^\Sigma = \sum_{i=1}^n \lvert v_i\rvert$ |
| Product | $\alpha_v^\Pi : [0,1]^n \to [-1,1]$ | $\alpha_v^\Pi(s) = \prod_{i:v_i=-1}(1-s_i) - \prod_{i:v_i=1}(1-s_i)$ | $\lambda_v^\Pi = \sum_{i=1}^n \lvert v_i\rvert$ |
| Top | $\alpha_v^{max} : [0,1]^n \to [-1,1]$ | $\alpha_v^{max}(s) = M_v(s) - M_{-v}(s)$, | $\lambda_v^{max} = \min\{2, \sum_{i=1}^n \lvert v_i\rvert\}$ |
| | | where $M_v(s) = \max\{0, v_1 \cdot s_1, \ldots, v_n \cdot s_n\}$, | |
| **Influence Functions** | | | |
| Linear($\kappa$) | $\iota_w^l : [-\kappa,\kappa] \to [0,1]$ | $\iota_w^l(s) = \mathrm{w} - \frac{\mathrm{w}}{\kappa} \cdot \max\{0,-s\} + \frac{1-\mathrm{w}}{\kappa} \cdot \max\{0,s\}$ | $\lambda_w^l = \frac{1}{\kappa}\max\{w, 1-w\}$ |
| Euler-based | $\iota_w^e : \mathbb{R} \to [\mathrm{w}^2, 1]$ | $\iota_w^e(s) = 1 - \frac{1-w^2}{1+w\cdot e^s}$ | $\lambda_w^e = \frac{1}{4}$ |
| p-Max($\kappa$) | $\iota_w^p : \mathbb{R} \to [0,1]$ | $\iota_\mathrm{w}^p = \mathrm{w} - \mathrm{w} \cdot h(-\frac{s}{\kappa}) + \mathrm{w} \cdot h(\frac{s}{\kappa})$ | $\lambda_w^p = \frac{p}{\kappa}\max\{w, 1-w\}$ |
| for $p \in \mathbb{N}$ | | where $h(x) = \frac{\max\{0,x\}^p}{1+\max\{0,x\}^p}$ | |

**Table 1: Some aggregation and influence functions with corresponding Lipschitz constants.**

| Semantics | Aggregation | Influence |
|---|---|---|
| DFQ($\kappa$) | Product | Linear($\kappa$) |
| Euler | Sum | Euler-based |
| QE($\kappa$) | Sum | 2-Max($\kappa$) |

**Table 2: Example semantics from the literature.**

but requires that the aggregation function yields values between $-1$ and 1. Hence, it cannot be combined with the sum-aggregation function. More generally, linear($\kappa$) requires that the aggregation function yields values between $-\kappa$ and $\kappa$. The Euler-based influence function has been used for the Euler-based semantics in [5]. It has some nice properties but causes an asymmetry between attack and support as we discuss later. The p-Max influence function avoids this asymmetry. The p-Max influence function with $p = 2$ is used for the quadratic energy model in [30]. By increasing the parameter $p$, we increase (decrease) the influence of aggregates larger (smaller) than 1. We add again a parameter $\kappa$ for the conservativeness.

Table 2 summarizes the building blocks of the DF-QuAD algorithm (DFQ), the Euler-based semantics (Euler) and the quadratic energy model (QE). We also add a conservativeness parameter to DFQ and QE.

## 3 CONVERGENCE AND OPEN-MINDEDNESS

As shown in [26], modular acceptability semantics always converge for acyclic graphs. The claim remains true for basic modular semantics. In fact, the limit can be computed in linear time by a single pass trough the graph as we explain in the following proposition.

Proposition 3.1 (Convergence and Complexity for Acyclic BAGs). *Let* $\mathrm{Deg}_S$ *be a basic modular semantics. For every acyclic BAG* $\boldsymbol{A} = (\mathcal{A}, \mathrm{w}, \mathcal{R}, \mathcal{S})$ *with n arguments, the limit*

$$\mathrm{Deg}_S(\boldsymbol{A}) = \lim_{k\to\infty} f_S^k(\mathrm{w}).$$

*exists and can be computed by the following algorithm:*

(1) *Compute a topological ordering of the arguments and set* $s^{(0)} \leftarrow \mathrm{w}$ *and* $k \leftarrow 1$.
(2) *Pick the next argument* $\mathcal{A}_i$ *in the order and set*

$$\mathrm{Deg}_S(\boldsymbol{A})_i = \iota_{\mathrm{w}_i}(\alpha_{g_i}(s^{(k-1)})).$$

(3) *Set* $k \leftarrow k+1$ *and repeat step 2 until* $k > n$.

*Provided that* $\alpha_g$ *and* $\iota_w$ *can be computed in linear time, the algorithm runs in linear time.*

We will now apply the contraction principle to unify and to generalize the convergence guarantees from [26]. A contraction is a Lipschitz-continuous function with Lipschitz-constant strictly smaller than 1. The contraction principle states intuitively that every contraction has a unique fixed-point that can be reached by applying the function repeatedly starting from an arbitrary point.

Lemma 3.2 (Contraction Principle). *If S is a complete metric space and if* $f : S \to S$ *is a contraction, then there exists one and only one* $x^* \in S$ *such that* $f(x^*) = x^*$*. In particular,* $\lim_{n\to\infty} f^n(x) = x^*$ *for all* $x \in S$.

A proof of the contraction principle can be found, for example, in [37]. The set $[0,1]^n$ of strength vectors with distance $d(x,y) = \|x - y\|$ defined by the maximum norm is indeed a complete metric space. Given a BAG with $n$ arguments such that $(\iota_{\mathrm{w}_i} \circ \alpha_{g_i})$ is a contraction for all $i = 1, \ldots, n$, the contraction principle guarantees that the strength values converge. As we will explain soon, the convergence results in [26] are special cases of the following result. In particular, we can relate convergence time to the Lipschitz-constants.

Proposition 3.3 (Convergence and Complexity for Contractive BAGs). *Let* $\boldsymbol{A}$ *be a BAG, let* $\mathrm{Deg}_S$ *be a basic modular semantics and let* $\lambda_{\boldsymbol{A},S} = \max_{1\le i\le n} \lambda_{g_i}^\alpha \cdot \lambda_{w_i}^\iota$*. If* $\lambda_{\boldsymbol{A},S} < 1$*, then the update function* $f_S$ *of* $\mathrm{Deg}_S$ *is a contraction with unique fixed point* $s^* = \mathrm{Deg}_S(\boldsymbol{A})$.

*Furthermore, for all* $\epsilon > 0$, $\|f_S^k(\mathrm{w}) - s^*\| \le \epsilon$ *for all* $k > \frac{\log \epsilon}{\log \lambda_{\boldsymbol{A},S}}$.

Note, in particular, that the convergence bound in the last line implies $\|f_S^k(\mathrm{w}) - s^*\| \le 10^{-n}$ for all $k > C \cdot n$, where C is a constant that decreases with the Lipschitz constants of the aggregation and influence functions. In this sense, the strength values converge in linear time. In order to relate Proposition 3.3 to the convergence results in [26], we briefly repeat them here.

Proposition 3.4 (Convergence Guarantees from [26]). *Consider a BAG* $\mathbf{A}$ *and a modular semantics that uses*

(1) *Sum for aggregation and an influence function whose derivative is strictly bounded by M. If the indegree of every argument in* $\mathbf{A}$ *is bounded by* $\frac{1}{M}$*, then* $f^n(x)$ *converges.*
(2) *Top for aggregation and an influence function whose derivative is strictly bounded by* $\frac{1}{2}$*. Then* $f^n(x)$ *converges.*

Both results are special cases of Proposition 3.3. For the first result, we can see from Table 1 that the Lipschitz-constant of sum-aggregation, when applied to a particular argument, corresponds to the indegree of the argument. That is, $\lambda^{\Sigma}_{g_i} = indegree(\mathcal{A}_i)$. Furthermore, if the derivative of a function is $B$, it is also Lipschitz-continuous with Lipschitz-constant $B$. Therefore, $\lambda^{\iota}_{w_i} < \frac{1}{M}$. Hence, if the maximal indegree in $\mathbf{A}$ is bounded by $M$, the condition of Proposition 3.3 becomes $\max_{1 \leq i \leq n} \lambda^{\Sigma}_{g_i} \cdot \lambda^{\iota}_{w_i} < \max_{1 \leq i \leq n} \frac{M}{M} = 1$ and is satisfied as well. For the second result, note from Table 1 that the Lipschitz-constant of top-aggregation can never be larger than 2. Hence, if the derivative of the influence function is bounded by $\frac{1}{2}$, the condition of Proposition 3.3 is satisfied as before.

Hence, Proposition 3.3 unifies the results from [26]. It is also more general and can immediately be applied to other aggregation functions like Product-aggregation. For the influence function, it is also slightly more general in the sense that bounded derivatives imply Lipschitz-continuity, but not the other way round. In many cases, practical influence functions will only be pointwise non-differentiable like Linear($\kappa$) or 1-Max($\kappa$). Proposition 3.3 still simplifies the investigation in these cases because we do not have to make any complicated case differentiations for such points. Proposition 3.3 implies several new convergence guarantees. We summarize some guarantees for product-aggregation in the following corollary.

Corollary 3.5. *Consider a BAG* $\mathbf{A}$ *with maximum indegree* $D = \max_{1 \leq i \leq n} indegree(\mathcal{A}_i)$*. When using a modular semantics with Product-aggregation, the strength values are guaranteed to converge*

- *if the Linear($\kappa$) influence function is used and* $D < \kappa$,
- *if the Euler-based influence function is used and* $D < \frac{\kappa}{4}$,
- *if the p-Max($\kappa$) influence function is used and* $D < \frac{\kappa}{p}$.

*When all weights in* $\mathbf{A}$ *are strictly between* 0 *and* 1*, then* $<$ *can be replaced with* $\leq$ *for Linear($\kappa$) and p-Max($\kappa$).*

*When using Sum-aggregation and p-Max($\kappa$), the strength values are guaranteed to converge if* $D < \frac{\kappa}{p}$*. Again,* $<$ *can be replaced with* $\leq$ *if all weights are strictly between* 0 *and* 1.

In order to show that these bounds cannot be improved much further, we give some tight examples based on a family of BAGs from [26]. We denote the members of the family by $\mathbf{A}(k, v_a, v_b)$. $\mathbf{A}(k, v_a, v_b)$ contains $k$ nodes $a_i$ with weight $v_a$ and $k$ nodes $b_i$ with weight $v_b$. All $a_i$ attack all $a_j$ and all $b_i$ attack all $b_j$ (including self-attacks). Furthermore, all $a_i$ support all $b_j$ and all $b_i$ support all $a_j$. Hence, the indegree of every argument in $\mathbf{A}(k, v_a, v_b)$ is $2k$ ($k$ supporters and $k$ attackers).

Figure 2 illustrates the behaviour of DFQ(1) and QE(1) for the BAG $\mathbf{A}(1, 0.9, 0.1)$, where the green and blue dots show the strength of argument $a_1$ and $b_1$ over a number of iterations. Both models start jumping between the same two states after a small number of iterations. Since $\mathbf{A}(1, 0.9, 0.1)$ has indegree 2, this is a tight example for DFQ(1) and QE(1) that shows that the general bounds given in Corollary 3.5 cannot be improved significantly.

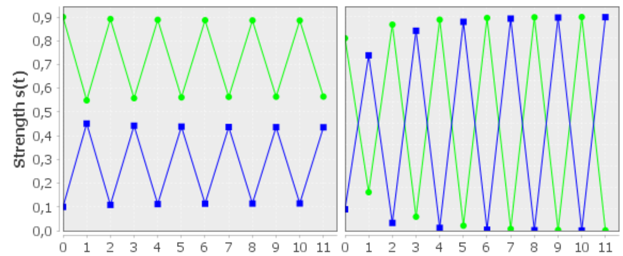


**Figure 2: Divergence of QE(1) (left) and DFQ(1) (right) for $\mathbf{A}(1, 0.9, 0.1)$.**

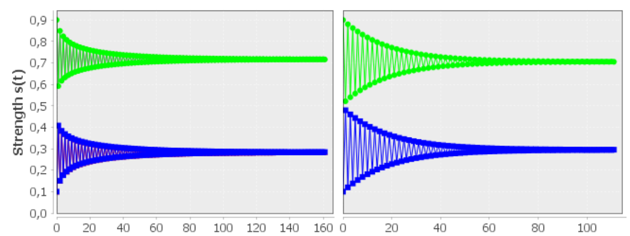


**Figure 3: Convergence of QE(2.1) (left) and DFQ(1.9) (right) for $\mathbf{A}(1, 0.9, 0.1)$.**

As we illustrate in Figure 3, we can solve the divergence problem by increasing the conservativeness parameter $\kappa$ of the semantics. Indeed, since increasing the conservativeness decreases the Lipschitz-constant, we can see from Proposition 3.3 that the convergence guarantees improve. However, of course, this also affects the semantics as we discuss next.

## Open-Mindedness

Proposition 3.4 implies that semantics that use top for aggregation and an influence function with derivative bounded from above strictly by $\frac{1}{2}$ are guaranteed to converge. Hence, when using the Euler-based influence function or influence functions that scale the influence of the aggregated value down by a constant $\kappa$ similar to Linear($\kappa$) and p-Max($\kappa$), the semantics converges in general. While this is a nice guarantee, it does not come without cost. The bound imposed on the growth of the influence function limits the semantics' ability to adapt the initial weight as we illustrate in the following example.

*Example 3.6.* Consider a BAG with one argument $a$ and $k$ arguments $b_i$ that attack $a$. All arguments have initial weight 0.9. Table 3 shows final strength values of argument $a$ for modular semantics with different building blocks. Naturally, when using top for aggregation, the final strength is independent of the number of attackers. We can also see that increasing the conservativeness parameter lets the final strength values keep closer to the initial weights. Note also that the Euler-based semantics is extremely conservative.

| $\alpha$ | $\iota$ | $k = 1$ | $k = 10$ | $k = 100$ |
|---|---|---|---|---|
| Sum | Euler | 0.862 | 0.811 | 0.811 |
| Top | Euler | 0.862 | 0.862 | 0.862 |
| Sum | 2-Max(1) | 0.498 | 0.012 | 0.001 |
| Top | 2-Max(1) | 0.498 | 0.498 | 0.498 |
| Sum | 2-Max(5) | 0.873 | 0.213 | 0.004 |
| Top | 2-Max(5) | 0.873 | 0.873 | 0.873 |

**Table 3: Strength values of $a$ under different semantics and increasing number of attackers $k$ for BAG from Example 3.6.**

Arguably, a semantics should be able to move the strength values arbitrarily close to the extreme values 0 or 1 if sufficient evidence against or for the argument is given. We call such a semantics open-minded.

*Definition 3.7 (Open-Mindedness).* We say that an influence function $\iota : [l, u] \rightarrow [0, 1]$ is open-minded if $\lim_{a \to l} \iota(a) = 0$ and $\lim_{a \to u} \iota(a) = 1$.

We call a basic modular semantics with aggregation function $\alpha : [0, 1]^n \rightarrow [l, u]$ open-minded when its influence function restricted to the domain $[l, u]$ is open-minded.

Note that we do not demand that the influence function ever yields the extreme values 0 or 1 (this would be in conflict with the Resilience axiom from [5]), we only demand that it is possible to get arbitrarily close to these bounds. For the Euler-based influence function, we have $\lim_{a \to -\infty} \iota^e_{\mathrm{w}}(a) = 1 - \frac{1-\mathrm{w}^2}{1+\mathrm{w} \cdot 0} = \mathrm{w}^2$. Hence, the Euler-based semantics is not open-minded since it does not admit final strength values smaller than $\mathrm{w}^2$. For example, in Table 3, the Euler-based influence function cannot yield a final strength value smaller than $0.9^2 = 0.81$. Linear($\kappa$) and p-Max($\kappa$) are open-minded influence functions and DFQ(1) and QE($\kappa$) are open-minded semantics. However, DFQ($\kappa$) is not open-minded for $\kappa > 1$. Also, none of the semantics with general convergence guarantees from [26] are open-minded. These negative results are all special cases of the following proposition.

Proposition 3.8. *Consider a basic modular semantics with aggregation function $\alpha : [0, 1]^n \rightarrow [-B, B]$ and influence function $\iota$ whose Lipschitz constant is bounded by $\lambda^\iota$. Then for every BAG $\boldsymbol{A} = (\mathcal{A}, \mathrm{w}, \mathcal{R}, \mathcal{S})$ with n arguments, the following bound is true for all $i = 1, \dots, n$:*

$$\mathrm{w}_i - B \cdot \lambda^\iota \leq \mathrm{Deg}_S(\boldsymbol{A})_i \leq \mathrm{w}_i + B \cdot \lambda^\iota.$$

For example, the Euler-based influence function has $\lambda^e = 0.25$. For aggregation with top, we have $B = 1$. Hence, when combining these two, no weight can change by more than 0.25.

It seems that when strong convergence guarantees can be derived from the contraction principle, they are bought at the expense of open-mindedness. The extreme case would be the constant influence function $\iota_{\mathrm{w}}(a) = \mathrm{w}$ that just assigns the initial weight to every aggregate. Its Lipschitz constant is 0 and every basic modular semantics that uses this influence function is guaranteed to converge trivially. As we let $\kappa$ in DFQ($\kappa$) and QE($\kappa$) go to infinity, we gradually increase our convergence guarantees, but simultaneously approach the constant influence function that leaves all weights unchanged. All currently known convergence guarantees for cyclic BAGs seem to be of this kind: we buy convergence guarantees at the expense of open-mindedness.

## 4 CONTINUOUS MODULAR SEMANTICS

We now look at another approach to improve convergence guarantees. Instead of making semantics more conservative, we will adapt the update approach. Roughly speaking, we will make updates more fine-grained. We will show that this approach leaves the semantics unchanged in cases where we have convergence guarantees. More importantly, it can still converge to a fixed-point of the semantics when the original updating approach diverges.

Roughly speaking, discrete update approaches work by applying an update formula to the initial weights repeatedly until the process converges. In case of basic modular semantics, the update formula is given by the function $(\iota_{\mathrm{w}_i} \circ \alpha_{g_i})$. In [30], it has been proposed to replace discrete models by continuous ones. Continuous models can be designed in a more descriptive way than discrete models. To this end, the continuous change of arguments' strength based on the strength of their attackers and supporters is described by means of differential equations. If the system of differential equations is designed carefully, it yields a unique solution $\sigma^{\mathbf{A}} : \mathbb{R}^+_0 \rightarrow \mathbb{R}^n$. Intuitively, the $i$-th component $\sigma^{\mathbf{A}}_i(t)$ tells us the strength of the $i$-th argument at (continuous) time $t$ and the final strength values correspond to the limit $\lim_{t \to \infty} \sigma^{\mathbf{A}}(t)$. Just like the limit $\lim_{k \to \infty} f^k_S(\mathrm{w})$ for discrete basic modular semantics may not exist, the limit $\lim_{t \to \infty} \sigma^{\mathbf{A}}(t)$ may not exist. However, if we can continuize a discrete model, the discrete model can actually be seen as a coarse approximation of the continuous model [30]. In particular, the continuous model may still converge when its discrete counterpart diverges as we will demonstrate soon. While there are currently no strong analytical guarantees for continuous models in cyclic BAGs, no divergence examples have been found either and experiments show that they can converge quickly for large cyclic BAGs [30]. Furthermore, sufficient conditions have been given under which discrete models can be continuized. As we will show next, the results can be generalized to all basic modular semantics.

Before stating the result, we add some explanations. The continuized model can be obtained as the unique solution of a system of differential equations. The equations basically describe how the strength evolves at each current point in time based on the current strength. This is done by defining the derivatives of the function $\sigma^{\mathbf{A}} : \mathbb{R}^+_0 \rightarrow \mathbb{R}^n$. As it turns out, in order to continuize a basic modular semantics, we can just define the derivative for the i-th strength value at time $t$ as the difference $(\iota_{\mathrm{w}_i} \circ \alpha_{g_i})(\sigma(t)) - \sigma_i(t)$. That is, as the difference between the result of applying the update function to the current state and the state itself. Note that the difference is 0 if $\sigma(t)$ is a fixed-point of the function $(\iota_{\mathrm{w}_i} \circ \alpha_{g_i})$. In this case, the strength value remains unchanged. If $(\iota_{\mathrm{w}_i} \circ \alpha_{g_i})(\sigma(t)) > \sigma_i(t)$, the difference, and hence the slope, will be positive and the strength value increases. This does again make intuitively sense because the strength will be shifted towards the strength value that is desired by the update formula. For the case $(\iota_{\mathrm{w}_i} \circ \alpha_{g_i})(\sigma(t)) < \sigma_i(t)$, the strength decreases symmetrically. We are now ready to state the

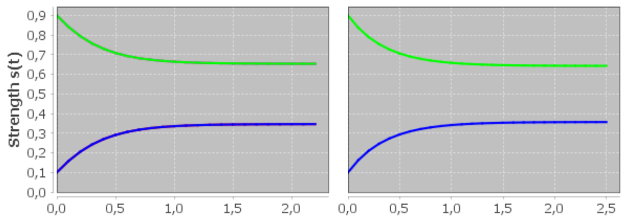


**Figure 4: Convergence of Continuous QE(1) (left) and Continuous DFQ(1) (right) for $\mathbf{A}(1, 0.9, 0.1)$.**

general result. As usual, we omit the function parameter $t$ when writing differential equations.

Proposition 4.1 (Continuizing Basic Modular Semantics). *Let $\mathrm{Deg}_S$ be a basic modular semantics with aggregation function $\alpha_g$ and influence function $\iota_w$.*

(1) *For all BAGs $\boldsymbol{A}$, the system of differential equations*

$$\frac{\mathrm{d}\sigma_i}{\mathrm{d}t} = (\iota_{\mathrm{w}_i} \circ \alpha_{g_i})(\sigma) - \sigma_i \tag{1}$$

*with initial conditions $\sigma_i(0) = \mathrm{w}(i)$ for $i = 1, \dots, n$ has a unique solution $\sigma^{\boldsymbol{A}} : \mathbb{R}_0^+ \to \mathbb{R}^n$.*

(2) *If $\sigma^{\boldsymbol{A}}$ converges and $s^* = \lim_{t\to\infty} \sigma^{\boldsymbol{A}}(t)$, then $s^*$ is a fixed-point of the update function $f_S$ of $\mathrm{Deg}_S$.*

(3) *If $\boldsymbol{A}$ is acyclic, the discrete and continuized models converge to the same limit.*

(4) *If $\sigma^{\boldsymbol{A}}$ converges and $f_S$ is a contraction, then the discrete and continuized models converge to the same limit.*

As opposed to the continuization result in [30], the proposition does not assume continuous differentiability of the update function and therefore applies to more general acceptability semantics like the DF-QuAD algorithm from [34] (DFQ(1) in Table 2).

We demonstrate in Figure 4 that continuizing discrete models can solve divergence problems. Whereas QE(1) and DFQ(1) diverged for $\mathbf{A}(1, 0.9, 0.1)$ (Figure 2), their continuized counterparts (Figure 4) converge. The intuitive reason for this is best explained by numerical solution techniques that approximate the continuous model $\sigma^{\mathbf{A}} : \mathbb{R}_0^+ \to \mathbb{R}^n$. The most naive technique is Euler's method. In our context, it initializes the strength values with the initial conditions given by the initial weights. That is, $\sigma^{\mathbf{A}}(0) = \mathrm{w}$. In order to compute $\sigma^{\mathbf{A}}(\delta)$ for some small $\delta > 0$, Euler's method uses a first-order Taylor approximation. The first order Taylor approximation of a differentiable function $f : \mathbb{R} \to \mathbb{R}^n$ about a point $t$ is given as $f_i(t + \delta) \approx f_i(t) + \delta \cdot \frac{\mathrm{d}f_i}{\mathrm{d}t}(t)$. Since we know $\sigma^{\mathbf{A}}(0) = \mathrm{w}$ and $\frac{\mathrm{d}\sigma_i^{\mathbf{A}}}{\mathrm{d}t}(0) = (\iota_{\mathrm{w}_i} \circ \alpha_{g_i})(\mathrm{w}) - \mathrm{w}$, the first-order Taylor approximation of $\sigma_i^{\mathbf{A}}(\delta)$ is $\mathrm{w} + \delta \cdot \big((\iota_{\mathrm{w}_i} \circ \alpha_{g_i})(\mathrm{w}) - \mathrm{w}\big)$. Having obtained our approximation for $\sigma^{\mathbf{A}}(\delta)$, we can move on approximating $\sigma^{\mathbf{A}}(2 \cdot \delta)$ analogously. In this way, we can approximate $\sigma^{\mathbf{A}}(t)$ for all $t > 0$ until the strength values converge. $\delta$ is called the step-size of the approximation and as $\delta \to 0$, the approximation error goes to 0 by differentiability of $\sigma^{\mathbf{A}}$.

Interestingly, the discrete update scheme turns out to be a Taylor-approximation with step size 1. To see this, just plug in $\delta = 1$. Then the approximation of $\sigma^{\mathbf{A}}(1)$ is $\mathrm{w} + 1 \cdot \big((\iota_{\mathrm{w}_i} \circ \alpha_{g_i})(\mathrm{w}) - \mathrm{w}\big) = (\iota_{\mathrm{w}_i} \circ \alpha_{g_i})(\mathrm{w})$. Notice that this is just our update formula applied to the initial weights once. Hence, applying the update formula once can be seen as a very coarse approximation of the continuous model at time 1 and, more generally, applying the update formula $k$ times can be seen as a coarse approximation of the continuous model at time $k$. Due to this coarseness, we may actually jump from the function graph of the true solution to the function graph of a solution for different initial conditions. This may cause divergence when the algorithm starts jumping back and forth between two function graphs. We can avoid these jumps by decreasing $\delta$. We illustrate this in Figure 5 for DFQ(1) and the BAG $\mathbf{A}(1, 0.9, 0.1)$. As we decrease $\delta$ from 1 to 0.8, the oscillations already become weaker, but the step size is not sufficiently small to avoid divergence. For $\delta = 0.5$, the oscillations die out and the true limit shown in Figure 4 is eventually reached.

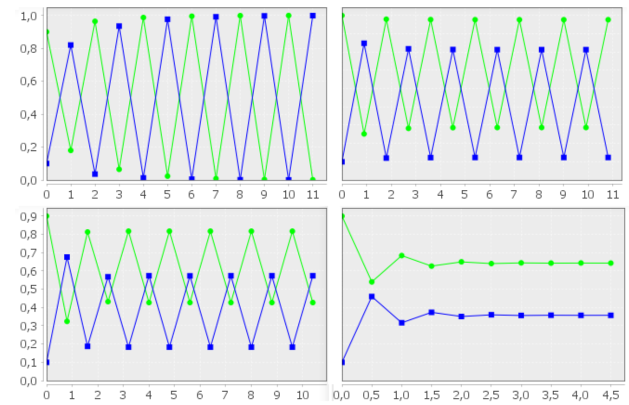

**Figure 5: Approximating Continuous DFQ(1) with Euler's method for $\mathbf{A}(1, 0.9, 0.1)$ with $\delta = 1$ (upper left), $\delta = 0.9$ (upper right), $\delta = 0.8$ (lower left) and $\delta = 0.5$ (lower right)**

Note that we refer to Euler's method only for didactic reasons. The results in Figure 4 were computed using the classical Runge-Kutta method RK4 that provides much stronger approximation guarantees [29].

## 5 DUALITY PROPERTY

In order to complement the semantical properties of basic modular semantics, we now generalize a symmetry property introduced in [30] to the setting from [26]. Intuitively, our symmetry property should assure that attackers move the strength from the initial weight towards 0 in the same way as supporters move the strength from the initial weight towards 1. This can be described by constraints on the aggregation and influence functions as follows.

*Definition 5.1 (Duality).* A basic modular semantics with aggregation function $\alpha_g$ and influence function $\iota_w$ satisfies Duality iff

(1) $\alpha_g(s) = -\alpha_{-g}(s)$ for all $s \in [0, 1]^n$ and
(2) $1 - \iota_{(1-\mathrm{w})}(a) = \iota_{\mathrm{w}}(-a)$ for all $\mathrm{w} \in [0, 1]$.

The aggregation condition says that when we switch the role of attackers and supporters (replace $g$ with $-g$), the aggregated strength value should just switch sign. For the special case $\mathrm{w} = 0.5$, the influence condition says that a positive aggregate must yield the

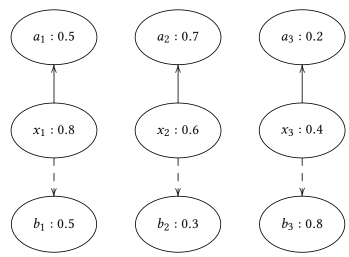


**Figure 6: Duality Example.**

| | $a_1$ | $b_1$ | $a_2$ | $b_2$ | $a_3$ | $b_3$ |
|---|---|---|---|---|---|---|
| Weight w | 0.50 | 0.50 | 0.70 | 0.30 | 0.20 | 0.80 |
| Euler | 0.39 | 0.65 | 0.63 | 0.41 | 0.15 | 0.84 |
| DFQ(1) | 0.10 | 0.90 | 0.28 | 0.72 | 0.12 | 0.88 |
| QE(1) | 0.30 | 0.70 | 0.51 | 0.49 | 0.17 | 0.83 |

**Table 4: Initial weight and strength values for arguments in Figure 6 under semantics from Table 2.**

same distance to 1 as the negative aggregate yields to 0. If $\mathrm{w} \neq 0.5$, there is a natural asymmetry because the initial weight is now either closer to 0 or 1. However, a negative aggregate for weight $w$ should still yield the same distance to 0 as the positive aggregate yields to 1 for weight $1 - w$. In the following proposition, we give a more intuitive interpretation of Duality.

Proposition 5.2. *Let* $\mathrm{Deg}_S$ *be a basic modular semantics that satisfies Duality and let* $\boldsymbol{A} = (\mathcal{A}, \mathrm{w}, \mathcal{R}, \mathcal{S})$ *be a BAG such that* $\mathrm{Deg}_S(\boldsymbol{A}) = s^* \neq \bot$. *If there are* $\mathcal{A}_i, \mathcal{A}_j$ *such that*

(1) $g_i = -g_j$ *or, more generally,* $\alpha_{g_i}(s^*) = -\alpha_{g_j}(s^*)$,
(2) $\mathrm{w}_i = 1 - \mathrm{w}_j$,

*then* $\mathrm{Deg}_S(\boldsymbol{A})_i = 1 - \mathrm{Deg}_S(\boldsymbol{A})_j$.

The basic case of the first condition says that $\mathcal{A}_i$'s attackers are $\mathcal{A}_j$'s supporters and vice versa. This is intuitive, but somewhat restrictive. The more general version says that the magnitude of the aggregated strength at $\mathcal{A}_i$ and $\mathcal{A}_j$ is equal, but it acts in different directions. The second condition says that the initial weights of $\mathcal{A}_i$ and $\mathcal{A}_j$ are complementary. Intuitively, we should then expect that their final strength values will also be complementary. We illustrate this in the following example.

*Example 5.3.* Consider the BAG in Figure 6. Table 4 shows the strength values for the three semantics from Table 2. The asymmetry of the Euler-based semantics can already be seen from the subgraph with indices 1. Whereas the support of $x_1$ increases the strength of $b_1$ by 0.15, its attack decreases the strength of $a_1$ only by 0.11. Both the DF-QuAD algorithm and the quadratic energy model induce a symmetrical impact for attacks and supports.

As we move the initial weight away from 0.5, there is a natural asymmetry caused by the fact that the distance from the initial weight to 0 and 1 is now different. However, attack and support should still behave in a dual manner. For the subgraph with indices 2, the initial weight of $a_2$ and $b_2$ is moved away from 0.5 by 0.2 in different directions. Again, the increase caused by a support should equal the decrease caused by an attack. For the DF-QuAD algorithm, the change is 0.42, for the quadratic energy model 0.19. Similarly, for the subgraph with indices 3, the DF-QuAD algorithm causes a change of 0.08, the quadratic energy model causes a change of 0.03.

In Table 1, all building blocks other than the Euler-based influence functions can be selected in order to satisfy duality as we show in the following proposition.

Proposition 5.4. *The Sum-, Product- and Top-aggregation functions satisfy condition 1 in Definition 5.1. The Linear(κ) and p-Max(κ) influence functions satisfy condition 2 in Definition 5.1.*

Since the DF-QuAD algorithm and the quadratic energy model are constructed from these building blocks, an immediate consequence is that they satisfy duality.

# 6 IMPLEMENTING MODULAR SEMANTICS WITH ATTRACTOR

The framework of modular semantics and has been implemented in the Java library Attractor[2] [31]. The user can initialize modular semantics with different combinations of aggregation and influence functions and can use existing implementations of algorithms to compute strength values using discrete (by using Euler's method with step size 1) or continuous semantics. Implementations of the aggregation and influence functions discussed here already exist, but new functions can be added easily by implementing existing interfaces. For example, the semantics of the DF-QuAD algorithm can be initialized with the following three lines of code:

```
AggregationFunction agg = new ProductAggregation();
InfluenceFunction inf = new LinearInfluence(1);
ContinuousModularModel mod =
    new ContinuousModularModel(agg, inf);
```

Attractor contains implementations of RK4 (for reliable computations) and Euler's method (for simulating discrete semantics and illustration purposes). Both implementations have a printing variant that automatically generates plots like in Figure 4 (RK4) and Figure 2 (Euler) while computing the solution. The plots are generated by JFreeChart[3]. For example, in order to use the plotting variant of RK4, we can add the following code:

```
AbstractIterativeApproximator approximator =
    new PlottingRK4(mod);
mod.setApproximator(approximator);
```

Finally, the strength values for a BAG can be computed. Attractor provides a simple syntax to define BAGs in text files. The file format is inspired by the format used in ConArg[4] [14], but adds weights and support relations. BAGs can also be defined programmatically if more flexibility is required. We refer to [31] for details on creating BAGs. Assuming that a BAG file is given, the strength values can

[2] https://sourceforge.net/projects/attractorproject
[3] http://www.jfree.org/jfreechart/
[4] http://www.dmi.unipg.it/conarg/

be computed by adding the following lines of code:

```
BAGFileUtils fileUtils = new BAGFileUtils();
BAG bag = fileUtils.readBAGFromFile(file);
mod.setBag(bag);
mod.approximateSolution(10e-2, 10e-4, true);
```

The two numerical parameters correspond to the step size and the termination condition, respectively. Mathematically, the algorithms converge to a fixed-point at which all derivatives will be 0. However, even mathematically, the fixed-point may not be reached in finite time. In practice, we also have to think about numerical accuracy, and so we usually stop when the derivatives are sufficiently small. Let us emphasize that the user does not have to think about derivatives. The derivatives are given by the differential equations. When adding new aggregation or influence functions, the differential equations are automatically derived as explained in Proposition 4.1. The logic is already implemented in the class *ContinuousModularModel*. So when implementing a new aggregation or influence function, only the logic for aggregating strength values or adapting the initial weight needs to be implemented.

## 7 RELATED WORK

In the original abstract argumentation framework [19], arguments can only be attacked by other arguments. *Bipolar argumentation frameworks* [6, 16, 27] add a support relation. Classical semantics can only accept or reject arguments [8], but various proposals have been made to allow for a more fine-grained evaluation. Among others, it has been suggested to apply tools from probabilistic reasoning [18, 20–23, 25, 28, 33, 35, 36] or to rank arguments based on fixed-point equations [12, 13, 17, 24] or the graph structure [2, 15].

In recent years, several weighted bipolar argumentation frameworks as considered here have been presented [5, 11, 26, 30, 34]. The QuAD algorithm from [11] was designed to evaluate the strength of answers in decision-support systems. However, it can show discontinuous behaviour that is undesirable in some cases. The DF-QuAD algorithm (Discontinuity-free QuAD) [34] was proposed as an alternative that avoids this behaviour. Some additional interesting semantical guarantees are given by the Euler-based semantics that was introduced in [5]. The QuAD algorithms mainly lack these properties due to the fact that their aggregated strength values *saturate*. That is, as soon, as an attacker (supporter) with strength 1 exists, the other attackers (supporters) become irrelevant for the aggregated value. The Euler-based semantics avoids many problems, but has some other drawbacks that can be undesirable. Arguments initialized with strength 0 or 1 remain necessarily unchanged under Euler-based semantics and, as we saw, attacks and supports have an asymmetrical impact. The *quadratic energy model* introduced in [30] avoids these problems. In [26], some other related models have been studied that use initial weights, an aggregation and an influence function as well, but the final strength values can also take values from the interval $[-1, 1]$ or general real numbers. Other aggregation and influence functions for these cases have been discussed in [26] as well.

A first collection of general axioms for weighted bipolar frameworks has been presented in [5]. Several authors noted recently that the axioms can be simplified by using more elementary properties [7, 9, 26]. The idea of modular semantics from [26] seems particularly useful because it allows creating new semantics with interesting guarantees by simply combining suitable aggregation and influence functions. This approach bears some resemblance to representation theorems considered in other fields that relate semantical properties of operators to elementary properties of functions that can be used to create these operators. Some ideas similar to modular semantics have been invented independently for the special case where only attack relations are present in [7].

## 8 DISCUSSION AND FUTURE WORK

We extended the framework of modular semantics from [26] in several directions. Our main focus was on convergence guarantees. We generalized the convergence guarantees from [26] to Lipschitz-continuous aggregation and influence functions. This allowed us, in particular, to derive convergence guarantees for semantics based on product-aggregation like the DF-QuAD algorithm. We also complemented the results from [26] with runtime guarantees based on the approximation accuracy and the Lipschitz constants. The Lipschitz constants provided in Table 1 can be used to derive further convergence guarantees in combination with Proposition 3.3. There are many other interesting candidates for aggregation and influence functions and, provided that they are Lipschitz-continuous, Proposition 3.3 can be applied to derive convergence guarantees easily. For example, truncated sums like the Lukasiewicz T-conorm could be interesting. In combination with the linear influence function they can guarantee that the extreme values 0 and 1 are taken in desirable cases (e.g., if there is only one attacker/supporter with strength 1) while avoiding the saturation property of the QUAD algorithms.

As we discussed, convergence guarantees for discrete models are often bought at the expense of open-mindedness. We demonstrated that we can avoid divergence problems without giving up open-mindedness by continuizing discrete models as proposed in [30]. It is currently an open question if and under which conditions continuous models converge for general cyclic BAGs, but until now, no divergence examples have been found. The continuization of all basic modular semantics yields a well-defined continuous model as Proposition 4.1 explains. The limits of discrete and continuized models are guaranteed to be equal for acyclic BAGs and for cyclic BAGs that induce a contractive update function. Further investigations are necessary, but it currently seems that whenever a discrete model converges, the continuized model converges to the same solution.

Semantically, we complemented modular semantics with the Duality property. After relating this property to elementary properties of aggregation and influence functions, it can be checked more easily. We showed, in particular, that it is satisfied by DF-QuAD.

Finally, we explained how weighted argumentation problems can be solved with the Java library Attractor. Modular semantics allow for very convenient abstractions. Dependent on the user's expertise, new semantics can be implemented completely from scratch, can be constructed from self-implemented aggregation and influence functions or by just combining pre-implemented aggregation and influence functions. A graphical user interface is work in progress.

## REFERENCES


[1] Teresa Alsinet, Josep Argelich, Ramón Béjar, Cèsar Fernández, Carles Mateu, and Jordi Planes. 2017. Weighted argumentation for analysis of discussions in Twitter. *International Journal of Approximate Reasoning* 85 (2017), 21–35.

[2] Leila Amgoud and Jonathan Ben-Naim. 2013. Ranking-based semantics for argumentation frameworks. In *International Conference on Scalable Uncertainty Management (SUM)*. Springer, 134–147.

[3] Leila Amgoud and Jonathan Ben-Naim. 2016. Axiomatic Foundations of Acceptability Semantics. In *International Conference on Principles of Knowledge Representation and Reasoning (KR)*. 2–11.

[4] Leila Amgoud and Jonathan Ben-Naim. 2016. Evaluation of arguments from support relations: Axioms and semantics. In *International Joint Conferences on Artificial Intelligence (IJCAI)*. pp–900.

[5] Leila Amgoud and Jonathan Ben-Naim. 2017. Evaluation of arguments in weighted bipolar graphs. In *European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU)*. Springer, 25–35.

[6] Leila Amgoud, Claudette Cayrol, and Marie-Christine Lagasquie-Schiex. 2004. On the bipolarity in argumentation frameworks. In *International Workshop on Non-Monotonic Reasoning (NMR)*, Vol. 4. 1–9.

[7] Leila Amgoud and Dragan Doder. 2018. Gradual Semantics for Weighted Graphs: An Unifying Approach. In *International Conference on Principles of Knowledge Representation and Reasoning (KR)*. 613–614.

[8] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. 2011. An introduction to argumentation semantics. *The Knowledge Engineering Review* 26, 4 (2011), 365–410.

[9] Pietro Baroni, Antonio Rago, and Francesca Toni. 2018. How many properties do we need for gradual argumentation?. In *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 1736–1743.

[10] Pietro Baroni, Marco Romano, Francesca Toni, Marco Aurisicchio, and Giorgio Bertanza. 2013. An argumentation-based approach for automatic evaluation of design debates. In *International Workshop on Computational Logic in Multi-Agent Systems*. Springer, 340–356.

[11] Pietro Baroni, Marco Romano, Francesca Toni, Marco Aurisicchio, and Giorgio Bertanza. 2015. Automatic evaluation of design alternatives with quantitative argumentation. *Argument & Computation* 6, 1 (2015), 24–49.

[12] Howard Barringer, Dov M Gabbay, and John Woods. 2012. Temporal, numerical and meta-level dynamics in argumentation networks. *Argument & Computation* 3, 2-3 (2012), 143–202.

[13] Philippe Besnard and Anthony Hunter. 2001. A logic-based theory of deductive arguments. *Artificial Intelligence* 128, 1-2 (2001), 203–235.

[14] Stefano Bistarelli, Fabio Rossi, and Francesco Santini. 2016. ConArg: A Tool for Classical and Weighted Argumentation. In *International Conference on Computational Models of Argument (COMMA)*. 463–464.

[15] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. 2005. Graduality in Argumentation. *Journal of Artificial Intelligence Research* 23 (2005), 245–297.

[16] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. 2013. Bipolarity in argumentation graphs: Towards a better understanding. *International Journal of Approximate Reasoning* 54, 7 (2013), 876–899.

[17] Marco Correia, Jorge Cruz, and Joao Leite. 2014. On the Efficient Implementation of Social Abstract Argumentation. In *European Conference on Artificial Intelligence (ECAI)*. 225–230.

[18] Dragan Doder and Stefan Woltran. 2014. Probabilistic argumentation frameworks–a logical approach. In *International Conference on Scalable Uncertainty Management (SUM)*. Springer, 134–147.

[19] Phan Minh Dung. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence* 77, 2 (1995), 321–357.

[20] Phan Minh Dung and Phan Minh Thang. 2010. Towards (probabilistic) argumentation for jury-based dispute resolution. *International Conference on Computational Models of Argument (COMMA)* 216 (2010), 171–182.

[21] Anthony Hunter. 2014. Probabilistic qualification of attack in abstract argumentation. *International Journal of Approximate Reasoning* 55, 2 (2014), 607–638.

[22] Anthony Hunter, Sylwia Polberg, and Nico Potyka. 2018. Updating Belief in Arguments in Epistemic Graphs. In *International Conference on Principles of Knowledge Representation and Reasoning (KR)*. 138–147.

[23] Anthony Hunter and Matthias Thimm. 2014. Probabilistic argumentation with incomplete information. In *European Conference on Artificial Intelligence (ECAI)*. IOS Press, 1033–1034.

[24] Joao Leite and Joao Martins. 2011. Social abstract argumentation. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, Vol. 11. 2287–2292.

[25] Hengfei Li, Nir Oren, and Timothy J Norman. 2011. Probabilistic argumentation frameworks. In *International Workshop on Theory and Applications of Formal Argumentation*. Springer, 1–16.

[26] Till Mossakowski and Fabian Neuhaus. 2018. Modular Semantics and Characteristics for Bipolar Weighted Argumentation Graphs. *arXiv preprint arXiv:1807.06685* (2018).

[27] Nir Oren and Timothy J. Norman. 2008. Semantics for Evidence-Based Argumentation. In *International Conference on Computational Models of Argument (COMMA)*, Vol. 172. IOS Press, 276–284.

[28] Sylwia Polberg and Dragan Doder. 2014. Probabilistic abstract dialectical frameworks. In *European Workshop on Logics in Artificial Intelligence*. Springer, 591–599.

[29] Andrei D Polyanin and Valentin F Zaitsev. 2017. *Handbook of ordinary differential equations*. Chapman and Hall/CRC.

[30] Nico Potyka. 2018. Continuous Dynamical Systems for Weighted Bipolar Argumentation. In *International Conference on Principles of Knowledge Representation and Reasoning (KR)*. 148–157.

[31] Nico Potyka. 2018. A Tutorial for Weighted Bipolar Argumentation with Continuous Dynamical Systems and the Java Library Attractor. In *International Workshop On Non-Monotonic Reasoning (NMR)*.

[32] Nico Potyka. 2019. Extending Modular Semantics for Bipolar Weighted Argumentation (Technical Report). (2019). https://doi.org/10.13140/RG.2.2.15107.14881

[33] Henry Prakken. 2018. Probabilistic Strength of Arguments with Structure. In *International Conference on Principles of Knowledge Representation and Reasoning (KR)*. 158–167.

[34] Antonio Rago, Francesca Toni, Marco Aurisicchio, and Pietro Baroni. 2016. Discontinuity-Free Decision Support with Quantitative Argumentation Debates. In *International Conference on Principles of Knowledge Representation and Reasoning (KR)*. 63–73.

[35] Tjitze Rienstra. 2012. Towards a probabilistic dung-style argumentation system. In *International Conference on Agreement Technologies (AT)*. 138–152.

[36] Tjitze Rienstra, Matthias Thimm, Beishui Liao, and Leendert van der Torre. 2018. Probabilistic Abstract Argumentation based on SCC Decomposability. In *International Conference on Principles of Knowledge Representation and Reasoning (KR)*. 168–177.

[37] Walter Rudin. 1976. *Principles of mathematical analysis*. Vol. 3. McGraw-hill New York.

[38] Marcus Thiel, Philipp Ludwig, Till Mossakowski, Fabian Neuhaus, and Andreas Nürnberger. 2017. Web-retrieval supported argument space exploration. In *ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR)*. ACM, 309–312.