

Abstract Argumentation Frameworks with Domain Assignments

Alexandros Vassiliades^{1,2}, Theodore Patkos², Giorgos Flouris², Antonis Bikakis³, Nick Bassiliades¹, Dimitris Plexousakis²

¹Aristotle University of Thessaloniki, School of Informatics, Hellas

²Foundation for Research and Technology, Institute of Computer Science, Hellas

³University College London, Department of Information Studies, UK

{valexande, nbassili}@csd.auth.gr, {patkos, fgeo, dp}@ics.forth.gr, a.bikakis@ucl.ac.uk

Abstract

Argumentative discourse rarely consists of opinions whose claims apply universally. As with logical statements, an argument applies to specific objects in the universe or relations among them, and may have exceptions. In this paper, we propose an argumentation formalism that allows associating arguments with a domain of application. Appropriate semantics are given, which formalise the notion of *partial argument acceptance*, i.e., the set of objects or relations that an argument can be applied to. We show that our proposal is in fact equivalent to the standard Argumentation Frameworks of Dung, but allows a more intuitive and compact expression of some core concepts of commonsense and non-monotonic reasoning, such as the scope of an argument, exceptions, relevance and others.

1 Introduction

The production and evaluation of argumentative discourse rarely consists of opinions whose claims can precisely be judged as either true or false. There are almost always shades of truthfulness, which make a statement hold in certain situations and not in others. Non-monotonic, case-based, context-aware, defeasible and other forms of reasoning are based, to a smaller or larger extent, on this need to specify the conditions under which a proposition holds or, similarly, to refine the scope within which it can be accounted true. This paper proposes a formal method to support such a refinement process using an argumentative formulation.

Both abstract [Dung, 1995] and structured argumentation frameworks [Besnard *et al.*, 2014] enable the revision of the acceptability status of arguments in the light of new evidence or stronger counter-arguments. There is a rich literature on different semantics for deciding the acceptance of arguments, even for evaluating their strength. Yet, these frameworks are challenged when one does not need to decide whether an argument is generally valid or not, but rather to argue on the specific cases where it should be accepted.

Consider the following argument taken from a typical example of commonsense reasoning: “*This fruit is an apple, and since apples are typically red, this fruit is red*”. The

claim is based on a premise about apples that constitutes common knowledge. Apparently, there are numerous exceptions to this knowledge, such as Granny Smiths, rotten apples etc; nevertheless, an attack on the original argument with the position “*This apple is a Granny Smith and Granny Smiths are green, therefore this apple is green*” should not be read as an attempt to invalidate the argument altogether, but rather to limit the domain of its applicability: apples may still be red in general, and the same argument should still be considered valid in most cases, but when apples belong to a given cultivar, things may be different. Apart from commonsense reasoning, a similar need can be found in other domains, as for instance in negotiation dialogues, especially for multi-party *quid pro quo* conversations, or in the legal domain, where the organisation of similar cases within the same argument is a typical practice for modeling legislation.

The representation of exceptional cases can be accommodated by non-monotonic logics, but not by argumentation frameworks, in which arguments are *atomic entities*. Thus, unless one creates one argument for each different element of the argument’s domain, argumentation frameworks cannot distinguish between an argument about a specific entity and one referring to a set of entities of the same type, and cannot accommodate a notion of “partial acceptance”, where an argument is acceptable for some entities only.

To address this, we introduce a novel argumentation framework called “*Abstract Argumentation Framework with Domain assignments*” (AAFD), in which arguments are abstract, but have a domain of application describing the cases that each argument can be applied to. As with other abstract frameworks, AAFD includes a binary attack relation, whose role however is different: instead of invalidating an argument or reducing its strength, it limits its domain of application.

In the rest of the paper, following a discussion of related work (Section 2), we define the semantics for AAFDs (Section 3), which amounts to determining the applicability of each argument (called *scope*) taking into account its domain of application and that of its attackers. We show how an AAFD can be mapped to Abstract Argumentation Frameworks (AAFs), establishing in this way a connection between AAFDs and several types of non-monotonic reasoning, and exploit this mapping to prove various properties of the AAFD semantics (Section 4). We then consider various methods to support the modeller in defining the domain of application of

arguments in a more intuitive and expressive manner, using predicates (Section 5), and conclude (Section 6).

2 Related Work

Formal argumentation employs a simple model to capture various types of non-monotonic and commonsense reasoning that cannot be easily captured by standard logics, such as reasoning about trust or preferences. Since Dung’s seminal paper on AAFs [Dung, 1995], various extensions have been proposed, enabling attacks to be directed towards attacks [Baroni *et al.*, 2011], adding support [Cayrol and Lagasque-Schiex, 2009], preference [Amgoud and Vesic, 2014] or other types of relations [Martínez *et al.*, 2006], adding weights to arguments [Amgoud *et al.*, 2017] or attacks [Dunne *et al.*, 2011], associating arguments with values [Bench-Capon, 2003], or imposing hierarchies on arguments [Modgil, 2009]. However, little attention has been paid to what we call the *scope* of an argument, i.e., the class of entities that an argument can be applied to, despite the fact that it has been studied in other related areas, such as discourse analysis [Sillince, 1995], linguistic argumentation [Zubizarreta, 1992] and political argumentation [Zarefsky, 2008].

A recently proposed extension allows associating arguments with *topics* [Budán *et al.*, 2020]. Topics and arguments are semantically interrelated and the acceptability of an argument depends on the semantic proximity of the arguments that defend it. An important difference with our work is that acceptability is defined in terms of sets of arguments that satisfy certain conditions, so each argument is either accepted or not; in AAFDs, arguments may be partially accepted, i.e., accepted only for a set of entities.

Structured argumentation frameworks provide ways to represent and reason with domain knowledge. For example, Hunter [2018] investigates various ways to support non-monotonic reasoning in deductive argumentation [Besnard and Hunter, 2008]. This enables constructing arguments based on the (possibly defeasible) domain knowledge expressed in the base logic. However, while some of these logics enable the representation of *general* knowledge using variables, arguments can only support literals as their conclusions. Moreover, the commitment to a specific logic restricts the types of knowledge that we can reason about, which was one of our motivations for basing our work on AAFs.

Non-monotonic inheritance networks [Horty *et al.*, 1990], which consist of nodes representing individuals or classes, and directed links, representing taxonomic relations, capture the idea of attacks that restrict the scope of arguments. However, they only enable reasoning with taxonomic relations and can be translated to AAFs [Dung and Son, 1995].

3 Definitions

3.1 Domain Assignments

In this paper, we equip arguments with a *domain of application*, which corresponds to the elements of a given *universe* that the argument applies to; for example, an argument about apples refers to a specific set of elements, namely apples, and its validity does not extend to other elements of the universe, e.g., other fruits. The domain of application can be viewed,

intuitively, as a way of breaking down an argument into a possibly infinite number of tiny pieces, one for each element of its domain. These pieces do not refer to a logical part of the argument, e.g., a premise, but rather to the elements that the argument (as a whole) applies to, which can be accepted or rejected independently from the rest. This breakdown allows an argument to be attacked only for a specific part of its domain, i.e., some of its pieces, as in the introductory example, where an argument about Granny Smith apples only attacks the argument about apples on the specific sub-domain of Granny Smith apples, without affecting its validity for other apples. This idea has the consequence that an attack does not necessarily invalidate an argument totally, and thus an argument can be partially accepted, i.e., accepted for only some of the elements of its domain.

To formalise these ideas, we use *domain assignments*, which are functions that associate each argument to a *prior* and *posterior* domain of application. The prior domain of application (or simply *domain*) represents the elements of the universe that the argument applies to. Attacks among arguments limit the applicability of the attacked argument, resulting in the argument’s posterior domain of application (or simply *scope*), which represents the elements for which the argument is accepted as valid. Formally:

Definition 1 (Domain Assignment) Consider a non-empty set U , called the universe, and a set of arguments \mathcal{A} . A domain assignment of \mathcal{A} to U is a function that maps each argument to a subset of U , i.e., $D^U : \mathcal{A} \mapsto 2^U$.

For simplicity, we fix some universe U , and omit reference to it in the following.

Recall that *Abstract Argumentation Frameworks (AAFs)* are defined in [Dung, 1995] as a pair consisting of a set of arguments and a set of attacks (binary relation) among them. Abstract Argumentation Frameworks with Domain Assignments are AAFs equipped with a domain assignment:

Definition 2 (AAFDs) An Abstract Argumentation Framework with Domain assignments, or AAFD for short, is a triple $\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle$, such that \mathcal{A} is a set of arguments, $\mathcal{R} \subseteq \mathcal{A}^2$ is a binary relation among arguments, and $\overline{D^U} : \mathcal{A} \mapsto 2^U$ is a domain assignment of \mathcal{A} to U , called the prior domain assignment (or simply domain).

3.2 Semantics for AAFDs: Intuition, Desiderata

In Abstract Argumentation Frameworks [Dung, 1995], semantics are defined through extensions, which are subsets of the original set of arguments that satisfy specific properties. To allow partial acceptance, our analysis needs to be more fine-grained, and should concern the arguments’ domains, rather than the arguments themselves. In particular, the semantics of an AAFD is defined as a domain assignment (the scope) that determines the subsets of the arguments’ domains that satisfy certain properties, giving rise to semantics analogous to the classical ones (admissible, complete, etc.).

Before defining formally how the semantics of an AAFD is computed, we provide some relevant desiderata, inspired by postulates and principles defined in other related contexts (e.g., [Baroni *et al.*, 2019; Dung *et al.*, 2010]).

The first requirement is about the relationship between the domain and the scope of the argument. In particular, we argue that the scope of an argument must always be a subset of its domain. This is reasonable, because the applicability of an argument is an inherent property of the argument, determined by the modeller based on its logical content; therefore, the argumentation framework (and its attacks) cannot increase it. This gives rise to the *domain capping constraint (DC)*.

Further, the domain of an argument should not be arbitrarily reduced. For every element “missing” from the domain of an argument, there should be a valid reason, i.e., some external attack. In other words, if a subset of the domain of an argument receives no attack, it should be part of the argument’s scope. This gives rise to the *scope maximality constraint (SM)*. Note that a corollary of (DC) and (SM) combined is that an argument receiving no attack should be fully accepted, i.e., its scope should be equal to its domain.

The third requirement ensures that arguments are consistent to each other with respect to their scope. For example, if a attacks b , and u is in the domain of both a , b , then u cannot be in the scope of both arguments. This is called the *scope consistency constraint (SC)*.

Another interesting intuition stems from considering elements that always appear “together” in the arguments’ domains. Such elements are, in a sense, “indistinguishable”, and there seems to be no reason to differentiate among them. In other words, elements that always appear together in the arguments’ domains, should also appear together in their scope. This is the *equal treatment constraint (ET)*.

These four constraints are formally expressed as follows:

$$\mathbf{DC} \quad \forall a \in \mathcal{A}, D^U(a) \subseteq \overline{D^U(a)}$$

$$\mathbf{SM} \quad \forall a \in \mathcal{A} \text{ and } \forall u \in \overline{D^U(a)} \setminus D^U(a), \\ \exists b \in \mathcal{A} \text{ such that } (b, a) \in \mathcal{R} \text{ and } u \in \overline{D^U(b)}$$

$$\mathbf{SC} \quad \forall a, b \in \mathcal{A} \text{ and } \forall u \in U, (a, b) \in \mathcal{R} \text{ implies that } u \notin D^U(a) \cap D^U(b)$$

$$\mathbf{ET} \quad \forall a \in \mathcal{A}, u_1 \in \overline{D^U(a)} \text{ if and only if } u_2 \in \overline{D^U(a)}, \text{ implies } \\ \forall a \in \mathcal{A}, u_1 \in D^U(a) \text{ if and only if } u_2 \in D^U(a)$$

3.3 Semantics for AAFDs: Formal Definitions

To define our semantics, we first introduce the notion of *limiting the domain of application*: given a set of arguments \mathcal{A} , and two domain assignments D_1^U, D_2^U for \mathcal{A} , we say that D_1^U *limits* D_2^U , denoted by $D_1^U \sqsubseteq D_2^U$, if and only if $D_1^U(a) \subseteq D_2^U(a)$ for all $a \in \mathcal{A}$. Informally, $D_1^U \sqsubseteq D_2^U$ when D_1^U assigns arguments to “smaller sets” compared to D_2^U .

In the following, we define various classes of scope, which give rise to the respective semantics. In all of the following definitions, we assume a fixed AAFD $\mathcal{F} = \langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle$, and some scope $D^U : \mathcal{A} \mapsto 2^U$.

We start with the notion of compliant scope:

Definition 3 (Compliant scope) *The scope D^U is compliant if and only if $D^U \sqsubseteq \overline{D^U}$.*

A scope that is compliant simply guarantees that no arbitrary assignments are made as described by the DC re-

quirement (see also Proposition 5). As an example, consider Figure 1, which shows a simple AAFD $\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle$, where $U = \{x, y, z\}$, $\mathcal{A} = \{a, b\}$, $\mathcal{R} = \{(a, b), (b, a)\}$ and $\overline{D^U}(a) = \{x, y\}$, $\overline{D^U}(b) = \{x, y, z\}$. Then, the scope $D_1^U(a) = \{x, y\}$, $D_1^U(b) = \{y, z\}$ is compliant.

Definition 4 (Conflict-free) *The scope D^U is conflict-free if and only if it is compliant, and for any $a, b \in \mathcal{A}$, $u \in U$, if $(a, b) \in \mathcal{R}$ then $u \notin D^U(a) \cap D^U(b)$.*

Conflict-freeness ensures that the SC requirement is satisfied (see Proposition 5). Returning to the example of Figure 1, we note that D_1^U is not conflict-free, but $D_2^U(a) = \{x\}$, $D_2^U(b) = \{y, z\}$ is. *Acceptability* is defined as follows:

Definition 5 (Acceptability) *For $a \in \mathcal{A}$, $u \in U$, such that $u \in \overline{D^U}(a)$, the pair (a, u) is acceptable with respect to D^U if and only if, whenever $(b, a) \in \mathcal{R}$ and $u \in \overline{D^U}(b)$, there exists some $c \in \mathcal{A}$ such that $(c, b) \in \mathcal{R}$ and $u \in D^U(c)$.*

An argument $a \in \mathcal{A}$ is called *fully acceptable with respect to D^U* iff (a, u) is acceptable with respect to D^U for all $u \in \overline{D^U}(a)$ and $\overline{D^U}(a) \neq \emptyset$. It is called *non-acceptable with respect to D^U* iff there is no $u \in \overline{D^U}(a)$ such that (a, u) is acceptable with respect to D^U . It is called *partially acceptable with respect to D^U* if it is neither fully acceptable, nor non-acceptable with respect to D^U . Using Definition 5, we can now define admissibility in AAFDs as follows:

Definition 6 (Admissible) *The scope D^U is admissible if and only if it is conflict-free, and (a, u) is acceptable with respect to D^U , for all $a \in \mathcal{A}$, $u \in D^U(a)$.*

Continuing the example of Figure 1, we note that the scope $D_3^U(a) = \{x\}$, $D_3^U(b) = \{z\}$ is admissible, as can be easily verified.

Definition 7 (Complete) *The scope D^U is complete if and only if it is admissible, and for any $a \in \mathcal{A}$, $u \in \overline{D^U}(a)$, if (a, u) is acceptable with respect to D^U , then $u \in D^U(a)$.*

In the AAFD of Figure 1, the scope $D_4^U(a) = \{x, y\}$, $D_4^U(b) = \{z\}$ is complete. Complete scopes satisfy the SM requirement (see Proposition 5).

Definition 8 (Grounded) *The scope D^U is grounded if and only if it is a minimal (with respect to \sqsubseteq) complete scope.*

Definition 9 (Preferred) *The scope D^U is preferred if and only if it is a maximal (with respect to \sqsubseteq) admissible scope.*

Definition 10 (Stable) *The scope D^U is a stable scope if and only if it is conflict-free, and for any $a \in \mathcal{A}$, $u \in U$, if $u \in \overline{D^U}(a) \setminus D^U(a)$, then there exists $b \in \mathcal{A}$ such that $(b, a) \in \mathcal{R}$ and $u \in D^U(b)$.*

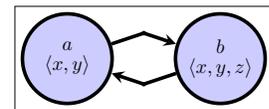


Figure 1. An example AAFD

To illustrate the above definitions, let us consider again the AAFD in Figure 1, and the following assignments:

- $D_5^U(a) = \{x, y\}, D_5^U(b) = \{z\}$.
- $D_6^U(a) = \{x\}, D_6^U(b) = \{y, z\}$.
- $D_7^U(a) = \emptyset, D_7^U(b) = \{z\}$.
- $D_8^U(a) = \emptyset, D_8^U(b) = \{x, y, z\}$.

We can easily show that all the above assignments are complete. The only grounded one is D_7^U , whereas D_5^U, D_6^U, D_8^U are preferred and stable.

None of the above types of scope captures the ideas behind the ET requirement described in Section 3.2. To formalise this intuition, we first define the concept of elements that always appear “together”:

Definition 11 (Indistinguishable) *Two elements $u_1, u_2 \in U$ are called indistinguishable, if and only if, for all $a \in \mathcal{A}$, $u_1 \in \overline{D^U}(a)$ if and only if $u_2 \in \overline{D^U}(a)$. We write $u_1 \sim u_2$ to denote that u_1, u_2 are indistinguishable.*

It is trivial to see that the relation \sim is an equivalence relation (i.e., reflexive, symmetric and transitive), and thus breaks down U into disjoint equivalence classes. This equivalence relation is used in the following definition:

Definition 12 (Canonical) *The scope D^U is canonical if and only if it is compliant, and for all $a \in \mathcal{A}$, $u_1, u_2 \in U$, whenever $u_1 \sim u_2$ and $u_1 \in D^U(a)$, it follows that $u_2 \in D^U(a)$.*

Looking at the previous examples, we note that D_4^U, D_5^U, D_7^U and D_8^U are canonical, whereas $D_1^U, D_2^U, D_3^U, D_6^U$ are not. Note that canonical semantics are orthogonal to other semantics, so any scope that is, e.g., admissible, complete, etc., may also be canonical.

In the following, we use the following shorthands for the semantics introduced above: **cm** for compliant, **cf** for conflict-free, **ad** for admissible, **co** for complete, **pr** for preferred, **gr** for grounded, **st** for stable and **ca** for canonical. We use σ as a catch-all symbol to indicate any of the above; for example, we write **co**-scope to refer to a complete scope, and σ -scope to refer to a scope of the type denoted by σ .

4 Results for AAFDs

4.1 Mapping AAFDs to AAFs

Interestingly, it can be shown that the semantics of AAFDs are fully compatible with the classical semantics of AAFs [Dung, 1995], i.e., an AAFD can be viewed as an AAF (and vice-versa, of course, e.g., when U is a singleton set). On the other hand, AAFDs provide a much more compact and intuitive representation. Below, we describe two different ways to map AAFDs to AAFs.

For the first mapping, we assume that each element in the domain of an argument is spawning an argument in the AAF (creating also attacks in the natural way). In other words, each element $u \in \overline{D^U}(a)$, along with the argument a itself (as a pair, a^u) are mapped into an argument in the AAF (see also Figure 2). Formally, the *natural mapping* is a mapping Φ_{nat} , which maps an AAFD $\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle$ to an AAF $\langle \mathcal{A}_*, \mathcal{R}_* \rangle = \Phi_{nat}(\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle)$, as follows:

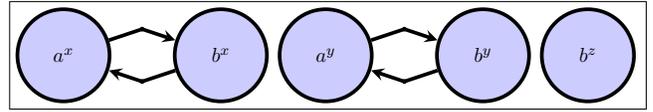


Figure 2. Applying Φ_{nat} on the AAFD of Figure 1

- $\mathcal{A}_* = \{a^u \mid a \in \mathcal{A}, u \in \overline{D^U}(a)\}$
- $\mathcal{R}_* = \{(a^u, b^u) \mid a^u, b^u \in \mathcal{A}_*, (a, b) \in \mathcal{R}\}$

For simplicity, we abuse notation, and denote by $\Phi_{nat}(D^U) = \{a^u \mid a \in \mathcal{A}, u \in D^U(a)\}$, where D^U is a compliant scope. In words, $\Phi_{nat}(D^U)$ returns all elements a^u such that u is in the scope of a (according to D^U).

The assumption for D^U being a compliant scope is crucial for defining $\Phi_{nat}(D^U)$. Indeed, if D^U was not compliant, then it would force us to spawn elements of the form a^u , where $u \notin \overline{D^U}(a)$; such elements do not appear in \mathcal{A}_* .

The following proposition shows the equivalence in the semantics of the two representations¹:

Proposition 1 *Take an AAFD $\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle$, the AAF $\langle \mathcal{A}_*, \mathcal{R}_* \rangle = \Phi_{nat}(\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle)$, and a compliant scope D^U in $\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle$. Then, for $\sigma \in \{\mathbf{cf}, \mathbf{ad}, \mathbf{co}, \mathbf{gr}, \mathbf{pr}, \mathbf{st}\}$, D^U is a σ -scope in $\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle$ if and only if $\Phi_{nat}(D^U)$ is a σ -extension in $\langle \mathcal{A}_*, \mathcal{R}_* \rangle$.*

Although Φ_{nat} achieves the required effect of mapping an AAFD into an AAF with equivalent semantics, it is not efficient because the size of the resulting AAF is highly dependent on the size of U , which is expected to be large (or even infinite). We show here a more sophisticated mapping which leads to an AAF whose size only depends on the number of arguments in AAFD, but not on the size of U . The idea stems from the notion of indistinguishable elements (Definition 11), and applies only for canonical scopes. To simplify presentation below, we denote by $\mathcal{C} = \{C_i\}$ the equivalence classes generated by \sim .

The *natural mapping with equivalences* $\Phi_{[nat]}$ is very similar to Φ_{nat} , except that it creates, for each $a \in \mathcal{A}$ and for each equivalence class C_i , just one argument in the AAF. In other words, if $a \in \mathcal{A}$, $u_1 \sim u_2$ (say $u_1, u_2 \in C_1$) and $u_1, u_2 \in \overline{D^U}(a)$, then, instead of creating two arguments in the AAF (namely a^{u_1}, a^{u_2}), $\Phi_{[nat]}$ creates just one (a^{C_1}). Figure 3 shows the application of $\Phi_{[nat]}$ on the AAFD of Example 1, for the equivalence classes $C_1 = \{x, z\}$, $C_2 = \{y\}$.

Formally, we define a mapping $\Phi_{[nat]}$, which maps an AAFD $\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle$ to an AAF $\langle \mathcal{A}_*, \mathcal{R}_* \rangle = \Phi_{[nat]}(\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle)$, as follows:

- $\mathcal{A}_* = \{a^C \mid a \in \mathcal{A}, C \in \mathcal{C}, C \cap \overline{D^U}(a) \neq \emptyset\}$
- $\mathcal{R}_* = \{(a^C, b^C) \mid a^C, b^C \in \mathcal{A}_*, C \in \mathcal{C}, (a, b) \in \mathcal{R}\}$

Again, we abuse notation, and denote by $\Phi_{[nat]}(D^U) = \{a^C \mid a \in \mathcal{A}, C \cap D^U(a) \neq \emptyset\}$, where D^U is a canonical scope. As before, this allows us to model the scope of AAFD arguments in the AAF representation.

¹All proofs are simple and omitted due to lack of space.

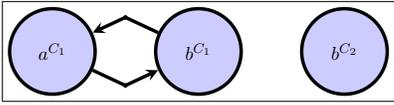


Figure 3. Applying $\Phi_{[nat]}$ on the AAFD of Figure 1

Note that the assumption for D^U being a canonical scope is crucial. Indeed, suppose that D^U is not canonical and take some $C \in \mathcal{C}$, $u_1, u_2 \in C$, such that $u_1 \in D^U(a)$, $u_2 \notin D^U(a)$. Then, it is not clear whether a^C (which essentially represents both u_1 and u_2 as candidates for inclusion in the scope of a) should be in $\Phi_{[nat]}(D^U)$.

The following result shows that $\Phi_{[nat]}$ does not mess with the semantics during the mapping of an AAFD to an AAF; note, however, that this result applies only when we restrict ourselves to the realm of canonical scopes:

Proposition 2 *Take an AAFD $\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle$, the AAF $\langle \mathcal{A}_*, \mathcal{R}_* \rangle = \Phi_{nat}(\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle)$, and a canonical scope D^U in $\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle$. Then, for $\sigma \in \{\text{cf}, \text{ad}, \text{co}, \text{gr}, \text{pr}, \text{st}\}$, D^U is a σ -scope in $\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle$ if and only if $\Phi_{[nat]}(D^U)$ is a σ -extension in $\langle \mathcal{A}_*, \mathcal{R}_* \rangle$.*

4.2 Complexity Results

We first address the problem of how to represent an AAFD. Let us consider an AAFD $\mathcal{F} = \langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle$. Set N_A, N_R, N_D, N_U the size of $\mathcal{A}, \mathcal{R}, \overline{D^U}, U$ respectively. Then: $N_R = O(N_A^2)$, $N_D = O(N_A N_U)$ and $|\mathcal{F}| = O(N_A + N_R + N_D) = O(N_A^2 + N_A N_U)$. For an AAF, $\mathcal{F}_A = \langle \mathcal{A}, \mathcal{R} \rangle$, the respective size is $|\mathcal{F}_A| = O(N_A + N_R) = O(N_A^2)$.

Another interesting consideration is the size of \mathcal{C} (say N_C). It can be shown that $N_C = O(2^{N_A})$. Indeed, suppose that $\mathcal{A} = \{a_1, \dots, a_{N_A}\}$, and set $S_i = \overline{D^U}(a_i)$. Then, each element $u \in U$ is related to a membership relation (\in or \notin) to each S_i , which gives a total of 2^{N_A} different combinations related to these membership relations. Moreover, $u_1 \sim u_2$ if and only if u_1, u_2 have the same membership relations to all S_i . Therefore, there are at most 2^{N_A} equivalence classes. Proposition 3 shows the worst-case size of an AAF that corresponds to an AAFD (under $\Phi_{nat}, \Phi_{[nat]}$):

Proposition 3 *Consider an AAFD $\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle$, such that $|\mathcal{A}| = N_A$, $|U| = N_U$. Set $\mathcal{F}_1 = \langle \mathcal{A}_1, \mathcal{R}_1 \rangle = \Phi_{nat}(\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle)$, $\mathcal{F}_2 = \langle \mathcal{A}_2, \mathcal{R}_2 \rangle = \Phi_{[nat]}(\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle)$. Then $|\mathcal{F}_1| = O(N_A^2 \cdot N_U^2)$, $|\mathcal{F}_2| = O(N_A^2 \cdot 2^{N_A})$.*

The most important consequence of Proposition 3 is that the size of $\Phi_{[nat]}(\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle)$ does not depend on the size of U . Thus, the most efficient representation depends on the relative size among \mathcal{A}, U . If we expect $|\mathcal{A}| \ll |U|$, as is usually the case, $\Phi_{[nat]}$ is more efficient, in terms of space, otherwise Φ_{nat} is more efficient. Also, the following corollary holds:

Corollary 1 *Take some $\mathcal{F} = \langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle$. If \mathcal{A}, U are finite, then $\Phi_{nat}(\mathcal{F})$ is finite. If \mathcal{A} is finite, then $\Phi_{[nat]}(\mathcal{F})$ is finite.*

Let us now consider time complexity. As implied by Proposition 3, we can map (in linear time, using Φ_{nat}) an

AAFD to an AAF of the same size and reason with that. Therefore, complexity of reasoning with an AAFD cannot be worse than the complexity of reasoning with an AAF of the same size. However, it cannot be better either, as can be observed by noticing that the AAFD, where $U = \{u\}$ and $\overline{D^U}(a) = \{u\}$ for all $a \in \mathcal{A}$, cannot be reasoned upon faster than the respective AAF (i.e., without the domains).

It should be noted that the above analysis regarding the space and time complexity applies for the simple representation of AAFDs, where domains are represented through the enumeration of their elements. More sophisticated representations (e.g., through predicates or other methods – see Section 5) may have a much smaller representational footprint, but may incur additional computational overheads (to determine the domains, as needed). Further analysis would require specific assumptions on the type of predicates/formulas used for the representation, and is omitted.

4.3 Properties of AAFD Semantics

The equivalence between the semantics of AAFDs and AAFs (Proposition 1) means that most of the results that have appeared in the literature regarding AAFs can be trivially applied to AAFDs as well (e.g., on the existence, multiplicity and relation of different types of scope). With regards to **cm** and **ca**-scopes, the following can be shown:

Proposition 4 *In any AAFD, there exists at least one **cm**-scope and at least one σ -scope that is also a **ca**-scope, for $\sigma \in \{\text{cf}, \text{ad}, \text{co}, \text{gr}, \text{pr}\}$. Also, if there exists a **st**-scope, then there exists a **st**-scope that is also a **ca**-scope.*

The following proposition shows how the various semantics defined in Section 3.3 meet the requirements described in Section 3.2:

Proposition 5 *Consider an AAFD $\langle \mathcal{A}, \mathcal{R}, \overline{D^U} \rangle$ and a scope D^U . Then:*

- For $\sigma \in \{\text{cm}, \text{cf}, \text{ad}, \text{co}, \text{pr}, \text{gr}, \text{st}\}$, if D^U is a σ -scope, then it satisfies **DC**.
- For $\sigma \in \{\text{co}, \text{pr}, \text{gr}, \text{st}\}$, if D^U is a σ -scope, then it satisfies **SM**.
- For $\sigma \in \{\text{cf}, \text{ad}, \text{co}, \text{pr}, \text{gr}, \text{st}\}$, if D^U is a σ -scope, then it satisfies **SC**.
- If D^U is a **ca**-scope, then it satisfies **ET**.

A corollary of Propositions 4, 5 is that a scope that is **ca-co**, **ca-gr**, or **ca-pr** always exists, and it satisfies all requirements set in Subsection 3.2. Thus, although suitability of semantics is application-dependent, we argue that **ca-co**, **ca-gr** and **ca-pr**-scopes seem more plausible for most applications. The same is true for **ca-st**-scopes, when they exist.

5 AAFDs with a Background Theory

The modelling method presented so far manages to deliver a more refined argumentation framework for reasoning about the domain and scope of arguments, in addition to the arguments themselves. In this section, we briefly explain how we can enhance the expressiveness of AAFDs by using predicates to describe the domain of arguments, rather than individuals. This can be important in several settings. Consider

for instance the following argument: “all doctors are licensed to practice general medicine”. Assume also that it so happens that in our universe all doctors are female, e.g., in the context of a specific hospital. If we represent the domain of this argument by spelling out the individuals, we lose the information that it is the profession of the individuals that this argument relies on, and not their gender.

A possible approach for lifting the expressive power of argumentation frameworks, especially abstract ones, is to rely on a background theory. This idea can be applied to AAFDs, in order to talk abstractly about attributes, categories, or classes of individuals when modelling the domain of arguments. According to the expressive power of the background theory, the resulting abstract framework can be structured to a lighter or heavier degree.

We define an AAFD enhanced with a simple background theory that relies on n -ary predicates to represent the domain of arguments, by adapting the definitions of Section 3. We assume an infinite set of n -ary predicate symbols \mathcal{P} , where n is fixed (i.e., for simplicity in presentation, we allow our universe to be described with unary, binary etc. predicates, but not with combinations of predicates having different arity). We use this set to define the predicate assignment $D^{\mathcal{P}}$ of a set of arguments A as follows:

Definition 13 (Predicate Assignment) *Given an infinite supply of predicate symbols \mathcal{P} of fixed arity n and a set of arguments A , a predicate assignment $D^{\mathcal{P}} : A \mapsto 2^{\mathcal{P}}$ is a function that maps each argument to a subset of \mathcal{P} .*

We define, in the usual manner, an interpretation \mathcal{I} that assigns meaning to predicates:

Definition 14 (Interpretation) *An interpretation \mathcal{I} that assigns meaning to the predicates in \mathcal{P} is a pair $\langle U, I \rangle$, where U is any nonempty set of objects, called the universe, and I is a mapping, called the interpretation mapping from the predicate symbols to relations over U , such that $I(P) \subseteq U^n$.*

For example, for the unary predicate symbol *Apple*, $I(\text{Apple})$ would be some subset of U , presumably the set of apples in that interpretation. We adjust Definition 1 so that domain assignments are described in terms of predicates:

Definition 15 (Domain Assignment, v2) *A domain assignment D^U of a set of arguments A over U is a mapping $D^U : A \mapsto 2^U$ such that: $D^U(a) = \bigcup_{P \in D^{\mathcal{P}}(a)} I(P)$.*

Finally, we revisit the definition of an AAFD (Definition 2) to take into account background knowledge. Intuitively, an argument talking about apples and pears is talking about any element in the universe of fruits that has either the property *Apple* or the property *Pear*. Formally:

Definition 16 (AAFDB) *An AAF with domain assignments and background theory, or AAFDB for short, is a quadruple $\langle A, \mathcal{R}, \overline{D^{\mathcal{P}}}, \mathcal{I} \rangle$, such that $\langle A, \mathcal{R} \rangle$ is an AAF, $\overline{D^{\mathcal{P}}} : A \mapsto 2^{\mathcal{P}}$ is the prior domain of A over \mathcal{P} , and \mathcal{I} an interpretation of the symbols in \mathcal{P} .*

We can then adapt the notions of acceptability, admissibility, completeness, etc. described in Section 3.1, to take advantage of the background theory, in order to define the semantics in an alternative way, using predicates; e.g., the following

definition of acceptability takes into consideration a context (set of predicates) within which attacks are considered:

Definition 17 (Acceptable, v2) *Given an AAFDB $\langle A, \mathcal{R}, \overline{D^{\mathcal{P}}}, \mathcal{I} \rangle$, an argument $a \in A$ and an element $u \in \bigcup_{P \in \overline{D^{\mathcal{P}}}(a)} I(P)$, the pair (a, u) is acceptable with respect to a predicate assignment $D^{\mathcal{P}}$ and a set of predicates $S \subseteq \mathcal{P}$ if and only if, $\forall b \in A$ such that $(b, a) \in \mathcal{R}$ and $u \in \bigcup_{P \in S \cap \overline{D^{\mathcal{P}}}(b)} I(P)$, $\exists c \in A$ such that $(c, b) \in \mathcal{R}$ and $u \in \bigcup_{P \in S \cap D^{\mathcal{P}}(c)} I(P)$.*

The idea is to focus only on attacks that concern specific predicates. Informally, an argument a is acceptable for an element of its domain u with respect to a predicate assignment $D^{\mathcal{P}}$ and a set of predicates S if, whenever there is an argument b that attacks a on u in the given context, there is another argument c that defends a on u . Note that attacks and defenses only consider elements that satisfy the predicates in S . This definition is similar to Definition 5, except that, here, the acceptability additionally takes into account the context of the framework, described as the set of predicates S .

One can further extend this approach with a more elaborate background theory for the domain of arguments that specifies, for instance, a taxonomy of concepts, or an ontological representation of the relations of concepts, or even a full first-order logic or a non-monotonic logic theory. The more expressive the background theory is, the more the framework will depart from its abstract notions and will approach the rationale of structured argumentation frameworks.

6 Conclusion and Future Work

We propose a novel argumentation framework, AAFD, where each argument has a domain of application that describes the entities that it refers to and can be applied on. This way, attacks essentially restrict an argument’s domain of application. This framework provides a formal computational method to support a refinement process using an argument-based formalisation that is rooted in argumentation theory, and can represent exceptions as arguments, essentially connecting the notions of classic argumentation (which supports atomic arguments) with non-monotonic reasoning.

For future work, we will define and implement algorithms for computing AAFD extensions, thereby exhibiting better average-case complexity compared to a naive approach mapping the AAFD to an AAF and reasoning using standard AAF algorithms. Moreover, an alternative mapping of AAFDs into AAFs based on a domain-based partitioning of arguments will be considered and explored. We will also further investigate the extension of AAFDs with a background theory, and develop AAFDs into dialectical frameworks, in order to use the expressiveness of AAFDs to approach commonsense reasoning problems.

Acknowledgments

This work has received funding from the Hellenic Foundation for Research and Innovation (HFRI) and the General Secretariat for Research and Technology (GSRT), under grant agreement No 188. We would like to thank Anthony Hunter for his valuable feedback on a previous version of this work.

References

- [Amgoud and Vesic, 2014] Leila Amgoud and Srdjan Vesic. Rich preference-based argumentation frameworks. *International Journal of Approximate Reasoning*, 55(2):585 – 606, 2014.
- [Amgoud *et al.*, 2017] Leila Amgoud, Jonathan Ben-Naim, Dragan Doder, and Srdjan Vesic. Acceptability Semantics for Weighted Argumentation Frameworks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI-17)*, page 56–62. AAAI Press, 2017.
- [Baroni *et al.*, 2011] Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Giovanni Guida. AFRA: Argumentation framework with recursive attacks. *International Journal of Approximate Reasoning*, 52(1):19–37, 2011.
- [Baroni *et al.*, 2019] Pietro Baroni, Antonio Rago, and Francesca Toni. From fine-grained properties to broad principles for gradual argumentation: A principled spectrum. *International Journal of Approximate Reasoning (IJAR)*, 105:252–286, 2019.
- [Bench-Capon, 2003] Trevor J. M. Bench-Capon. Persuasion in Practical Argument Using Value-based Argumentation Frameworks. *Journal of Logic and Computation*, 13(3):429–448, 06 2003.
- [Besnard and Hunter, 2008] Philippe Besnard and Anthony Hunter. *Elements of Argumentation*. MIT Press, 2008.
- [Besnard *et al.*, 2014] Philippe Besnard, Alejandro Garcia, Anthony Hunter, Sanjay Modgil, Henry Prakken, Guillermo Simari, and Francesca Toni. Introduction to structured argumentation. *Argument & Computation*, 5(1):1–4, 2014.
- [Budán *et al.*, 2020] Maximiliano C.D. Budán, Maria Laura Cobo, Diego C. Martinez, and Guillermo R. Simari. Proximity semantics for topic-based abstract argumentation. *Information Sciences*, 508:135 – 153, 2020.
- [Cayrol and Lagasque-Schiex, 2009] Claudette Cayrol and Marie-Christine Lagasque-Schiex. Bipolar abstract argumentation systems. In Guillermo Ricardo Simari and Iyad Rahwan, editors, *Argumentation in Artificial Intelligence*, pages 65–84. Springer, 2009.
- [Dung and Son, 1995] Phan Minh Dung and Tran Cao Son. Nonmonotonic inheritance, argumentation and logic programming. In V. Wiktor Marek, Anil Nerode, and M. Truszczynski, editors, *Logic Programming and Non-monotonic Reasoning*, pages 316–329, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.
- [Dung *et al.*, 2010] Phan Minh Dung, Francesca Toni, and Paolo Mancarella. Some design guidelines for practical argumentation systems. In Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Guillermo Ricardo Simari, editors, *Computational Models of Argument: Proceedings of COMMA 2010, Desenzano del Garda, Italy, September 8-10, 2010*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, pages 183–194. IOS Press, 2010.
- [Dung, 1995] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, September 1995.
- [Dunne *et al.*, 2011] Paul E. Dunne, Anthony Hunter, Peter McBurney, Simon Parsons, and Michael Wooldridge. Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artificial Intelligence*, 175(2):457 – 486, 2011.
- [Horty *et al.*, 1990] John F. Horty, Richmond H. Thomason, and David S. Touretzky. A skeptical theory of inheritance in nonmonotonic semantic networks. *Artificial Intelligence*, 42(2):311 – 348, 1990.
- [Hunter, 2018] Anthony Hunter. Non-monotonic Reasoning in Deductive Argumentation. *CoRR*, abs/1809.00858, 2018.
- [Martínez *et al.*, 2006] Diego C. Martínez, Alejandro J. García, and Guillermo R. Simari. On Acceptability in Abstract Argumentation Frameworks with an Extended Defeat Relation. In *Proceedings of the 2006 Conference on Computational Models of Argument: Proceedings of COMMA 2006*, page 273–278, NLD, 2006. IOS Press.
- [Modgil, 2009] Sanjay Modgil. Reasoning about preferences in argumentation frameworks. *Artificial intelligence*, 173(9-10):901–934, 2009.
- [Sillince, 1995] John AA Sillince. Shifts in focus and scope during argumentation. *Journal of Pragmatics*, 24(4):413–431, 1995.
- [Zarefsky, 2008] David Zarefsky. Strategic maneuvering in political argumentation. *Argumentation*, 22(3):317, 2008.
- [Zubizarreta, 1992] Maria Luisa Zubizarreta. The lexical encoding of scope relations among arguments. In *Syntax and the Lexicon*, pages 211–258. Brill, 1992.