Abstract Argumentation with Structured Arguments

Martin Baláž and **Jozef Frtús**
Department of Applied Informatics
Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava
balaz@ii.fmph.uniba.sk frtus@fmph.uniba.sk

Abstract

Argumentation is a process by means of which we reach acceptable conclusions through logical reasoning. To specify accepted arguments, abstract argumentation frameworks use only two notions - a set of abstract arguments and a defeat relation between arguments. To instantiate an abstract argumentation framework, further steps are needed to take.

We introduce abstract argumentation framework with structured arguments. Arguments are still abstract in the sense that they are not constructed but they are partially ordered with subargument relation. An attack represents direct conflict between two arguments. Defeat relation can be viewed as a successful attack propagated upwards through superarguments.

There exist many ways how to define which attacks are successful and under which conditions they propagate through superarguments. In this paper, we study properties and consequences of such design decisions. We analyze several existing approaches and show how they instantiate our framework.

Introduction

To describe the structure of a specific argumentation formalism, five steps are needed to take. Argumentation formalisms are based on underlying *logical language*, notion of *argument* and *attack* between arguments, *defeat* relation among arguments and a definition of the *status* of an argument (Prakken and Vreeswijk 2002). Each from these notions can be expressed by means of the previous. Status of an argument depends on the notions of argument and defeat relation, defeat relation depends on the structure of arguments and attack relation, attack relation depends on the notion of argument and underlying logical language, and the structure of arguments depends on underlying logical language.

Abstract argumentation frameworks (Dung 1995) are used to define the status of an argument only in terms of arguments and defeat relation. They neither consider the structure of arguments nor specify defeat relation. One of the most accepted semantics of abstract argumentation frameworks are preferred, stable, grounded, and complete extensions (Dung 1995).

In the existing literature, it is usually distinguished between attack and defeat relation. An attack represents direct conflict between arguments. In general, not all attacks are required to be successfull. For example, if two arguments rebut each other but one of them is more preferred than the other one, only one attack becomes successfull. (Prakken and Sartor 1997; Prakken 2010; García and Simari 2004) A defeat can be viewed as a successfull attack propagated upwards through superarguments. Similarly as attack does not always become successfull, defeat does not always have to propagate. For example, if an argument A rebuts an argument B, we may want to stop defeating a superargument of B which attacks an assumption of A. (Prakken and Sartor 1997)

In this paper, we introduce structured argumentation framework. We will use it to study various existing approaches how to define defeat relation in terms of structure of arguments and attack relation (Prakken and Sartor 1997; García and Simari 2004; Amgoud et al. 2005; Caminada and Amgoud 2007; Prakken 2010). We should clarify what we mean by "structured argument". Although one could require to distinguish at least between premises and conclusions of an argument, for our purposes in this paper, subargument relation is the only necessary part of argument's structure to be considered. Therefore we neither consider any specific structure on arguments nor specify attack relation. We only assume that arguments are partially ordered by subargument relation.

We also do not define specific defeat relation for a given attack relation. The main result of this work is an analysis of sufficient properties of structured argumentation frameworks such that standard Dung's extensions will be closed under subargument relation and consistent with respect to attack relation. An interesting result is that in the case of standard Dung's semantics, the propagation of the defeat relation on the left side does not matter.

Preliminaries

An *abstract argumentation framework* is a pair $(\mathcal{A}, \mathcal{D})$ where $\mathcal{A}$ is a set of *arguments* and $\mathcal{D} \subseteq \mathcal{A} \times \mathcal{A}$ is a *defeat* relation. An argument A *defeats* an argument B if $(A, B) \in \mathcal{D}$. A set of arguments S *defeats* an argument B if there exists an argument in S defeating B. A set of arguments S is *conflict-free* if there are no arguments A and B in S such that A defeats B. An argument A is *acceptable* with respect to a set of arguments S if each argument

B defeating A is defeated by S.

A conflict-free set of arguments S is *admissible* if each argument in S is acceptable with respect to S. A *preferred extension* is a maximal (with respect to set inclusion) admissible set of arguments. A conflict-free set of arguments S is called a *stable extension* if S defeats each argument which does not belong to S. The *characteristic function* $F\colon 2^{\mathcal{A}} \mapsto 2^{\mathcal{A}}$ is defined as $F(S) = \{A \in \mathcal{A} \mid A \text{ is acceptable with respect to } S\}$. The *grounded extension* is the least fixed point of F. An admissible set of arguments S is called *complete extension* if S contains all arguments acceptable with respect to S.

Structured Argumentation Frameworks

Now we are going to introduce structure on arguments by means of subargument relation. Superarguments are built upon their subarguments. Each conflict between two arguments has its origin. We will denote those sources of conflicts by attack relation. Each successful attack becomes defeat and under some conditions, it will propagate upwards through superarguments of conflicting arguments.

First we will give an informal example for illustration of subargument concept.

Example 1. $A =$ "Tweety flies because it is a bird" is an argument based on its subargument $B =$ "Tweety is a bird". In such cases we will write $B \sqsubseteq A$.

Definition 1. A *structured argumentation framework* is a triple $(\mathcal{A}, \sqsubseteq, \mathcal{R})$ where $(\mathcal{A}, \sqsubseteq)$ is a partially ordered set of *arguments* and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is an *attack* relation.

We say that an argument A is a *subargument of* an argument B if $A \sqsubseteq B$. A subargument A of an argument B is *strict* (denoted $A \sqsubset B$) if $A \neq B$. A set of arguments S is *closed* (under the subargument relation) if S contains all subarguments of each argument in S. A set of arguments S is *consistent* if there are no arguments A and B in S such that A attacks B.

In (Martínez, García, and Simari 2006), authors define similar concept as structured argumentation framework. They use conflict between arguments and a preference relation on arguments instead of attack between arguments. Attack relation can be viewed as resolution of a conflict with respect to preference relation. To keep definitions as simple as possible, we assume that preference relation is already contained in attack relation.

Although structured argumentation framework can be viewed as an instantiation of Dung's abstract argumentation framework, they are still abstract. They do not specify how to build arguments from their subarguments or how to define an attack between two arguments.

There exist many different ways how to instantiate an abstract argumentation framework $(\mathcal{A}, \mathcal{D})$ with a structured argumentation frameworks $(\mathcal{A}, \sqsubseteq, \mathcal{R})$, or to define a defeat relation $\mathcal{D}$ by means of a given set of arguments $(\mathcal{A}, \sqsubseteq)$ and an attack relation $\mathcal{R}$. In the following example we show that, in general, we can not simply let $\mathcal{D} = \mathcal{R}$.

Example 2. Let $\mathcal{A} = \{A, B, C\}$, $B \sqsubseteq C$ and $(A, B) \in \mathcal{R}$. If $\mathcal{D}_1 = \mathcal{R}$ then $E_1 = \{A, C\}$ is the grounded extension of $AF_1 = (\mathcal{A}, \mathcal{D}_1)$. We can see (Figure 1) that C belongs to the extension E_1 but its subargument B does not. This is very unintuitive, since the argument C is justified on the ungrounded base B. On the other hand, if we take defeat relation $\mathcal{D}_2 = \mathcal{R} \cup \{(A, C)\}$, $E_2 = \{A\}$ is the grounded extension of $AF_2 = (\mathcal{A}, \mathcal{D}_2)$ and it is closed under subargument relation. Closure under subargument relation is in the literature known as a *compositionality principle* (Prakken and Vreeswijk 2002).

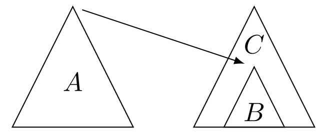

Figure 1: Visualization of the argumentation framework from the Example 2.

In general, we do not require that a defeat relation always propagates upwards through superarguments on the right side. But if it does not, backward defeat can be the only reason.

Property 1. *If an argument A defeats a subargument of an argument B then A defeats B or B defeats A.*

In the following figures, arguments are visualized as triangles, defeats as solid and attacks as dashed arrows.

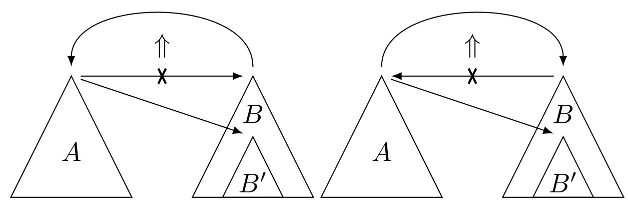

Figure 2: Consequences of the Property 1

Similarly, we may propagate defeat relation upwards through superarguments on the left side. As we have seen in the Example 2, propagation on the right side is necessary.

Property 2. *If a subargument of an argument A defeats a subargument of an argument B then A defeats B or B defeats A.*

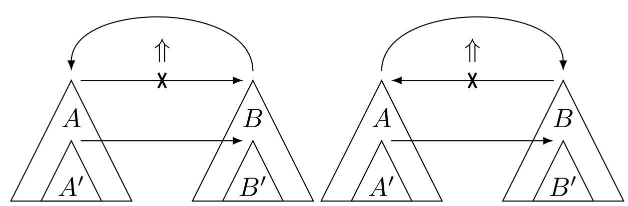

Figure 3: Consequences of the Property 2

Observation 1. *Let $(\mathcal{A}, \sqsubseteq, \mathcal{R})$ be a structured argumentation framework and $AF = (\mathcal{A}, \mathcal{D})$ be an abstract argumentation framework. If AF satisfies the Property 2 then AF also satisfies the Property 1.*

Proof. Let AF satisfies the Property 2. If we take $A' = A$, AF also satisfies the Property 1. □

It can be easily seen from Figure 3 that converse implication does not hold: argument A' defeats B', $B' \sqsubseteq B$, but neither A' defeats B, nor B defeats A'.

Lemma 1. *Let $(\mathcal{A}, \sqsubseteq, \mathcal{R})$ be a structured argumentation framework and $AF = (\mathcal{A}, \mathcal{D})$ be an abstract argumentation framework. Let S be a complete extension of AF. If each subargument of an argument A is acceptable with respect to S whenever A is acceptable with respect to S then S is closed.*

Proof. Let S be a complete extension of AF and A be an argument in S. Since S is complete, A is acceptable with respect to S. Then each subargument of A is acceptable with respect to S. Since S is complete, each subargument of S belongs to S. Thus S is closed. □

Proposition 1. *Let $(\mathcal{A}, \sqsubseteq, \mathcal{R})$ be a structured argumentation framework and $AF = (\mathcal{A}, \mathcal{D})$ be an abstract argumentation framework. If AF satisfies the Property 1 then each complete, preferred, stable, and grounded extension of AF is closed.*

Proof. Let S be a complete extension of AF, A be an argument acceptable with respect to S. Let A' be a subargument of A and B be an argument defeating A'. If B defeats A then S defeats B since A is acceptable with respect to S. Let B does not defeat A. According to the Property 1, A defeats B. Because A belongs to S, S defeats B. Then A' is acceptable with respect to S and according to the lemma 1, S is closed. Since preferred, stable and grounded extensions are complete extensions, they are also closed. □

Now we need to relate attack and defeat. The following two properties formalize the necessary condition for defeat relation. These properties indeed say that each defeat has its origin in attack.

Property 3. *If an argument A defeats an argument B then there exist subarguments A' of A and B' of B such that A' attacks B' and each argument between A' and A defeats each argument between B' and B.*

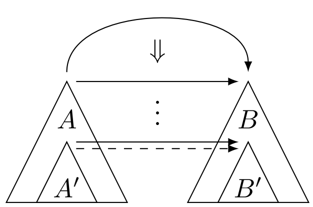

Figure 4: Visualization of the Property 3.

Similarly as we extended propagation of the defeat relation on the left side, we can restrict existence of the attack relation only to the right side.

Property 4. *If an argument A defeats an argument B then there exists a subargument B' of B such that A attacks B' and A defeats each argument between B' and B.*

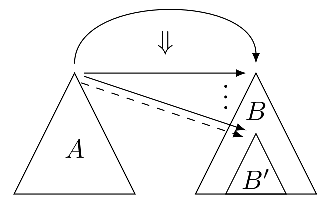

Figure 5: Visualization of the Property 4.

Observation 2. *Let $(\mathcal{A}, \sqsubseteq, \mathcal{R})$ be a structured argumentation framework and $AF = (\mathcal{A}, \mathcal{D})$ be an abstract argumentation framework. If AF satisfies the Property 4 then AF also satisfies the Property 3.*

Proof. Let AF satisfies the Property 4. If we take $A' = A$, AF also satisfies the Property 3. □

It can be easily seen from Figure 4 that converse implication does not hold, if $A' \sqsubset A$: argument A defeats B, $B' \sqsubseteq B$, but A does not attack B'.

Now it is time to show that in the case of standard Dung's semantics, the propagation of the defeat relation on the left side does not matter.

Definition 2. Let $(\mathcal{A}, \sqsubseteq, \mathcal{R})$ be a structured argumentation framework and $AF = (\mathcal{A}, \mathcal{D})$ be an abstract argumentation framework. A *left-restriction of* the defeat relation $\mathcal{D}$ to the attack relation $\mathcal{R}$ is a binary relation $\{(A, B) \in \mathcal{D} \mid \exists B' \sqsubseteq B \colon (A, B') \in \mathcal{R}\}$.

An attack is successful if it becomes defeat. Two abstract argumentation frameworks are right-equivalent if the propagation of successful attacks upwards through superarguments on the right side is the same in both frameworks.

Definition 3. Let $(\mathcal{A}, \sqsubseteq, \mathcal{R})$ be a structured argumentation framework, $AF_1 = (\mathcal{A}, \mathcal{D}_1)$ and $AF_2 = (\mathcal{A}, \mathcal{D}_2)$ be abstract argumentation frameworks. We say that AF_1 and AF_2 are *right-equivalent* if the left-restriction of $\mathcal{D}_1$ to $\mathcal{R}$ equals to the left-restriction of $\mathcal{D}_2$ to $\mathcal{R}$.

Example 3. Let $\mathcal{A} = \{A, A', B, B'\}$, $A' \sqsubset A$, $B' \sqsubset B$ and $\mathcal{R} = \{(B', A')\}$ (see Figure 6). Let $\mathcal{D}_1 = \mathcal{R}$, $\mathcal{D}_2 = \mathcal{D}_1 \cup \{(B', A)\}$ and $\mathcal{D}_3 = \mathcal{D}_2 \cup \{(B, A'), (B, A)\}$. Then argumentation frameworks $(\mathcal{A}, \mathcal{D}_2)$ and $(\mathcal{A}, \mathcal{D}_3)$ are right-equivalent, but $(\mathcal{A}, \mathcal{D}_1)$ and $(\mathcal{A}, \mathcal{D}_2)$ are not because $(B', A) \notin \mathcal{D}_1$.

Lemma 2. *Let $(\mathcal{A}, \sqsubseteq, \mathcal{R})$ be a structured argumentation framework, $AF_1 = (\mathcal{A}, \mathcal{D}_1)$ and $AF_2 = (\mathcal{A}, \mathcal{D}_2)$ be abstract argumentation frameworks. Let S be a complete extension of AF_1. If AF_1 and AF_2 are right-equivalent and satisfy the Properties 1 and 3 then*

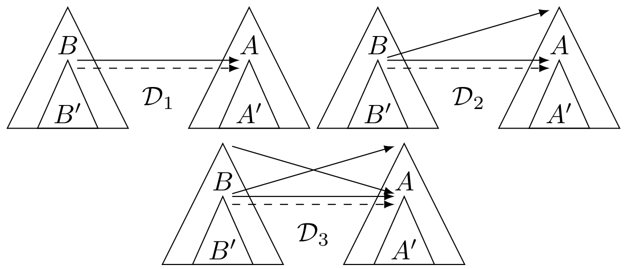

Figure 6: Visualization of argumentation frameworks form the Example 3.

1. *If S defeats an argument A in AF_1 then S defeats A in AF_2.*
2. *S is conflict-free in AF_2.*
3. *If an argument A is acceptable with respect to S in AF_1 then A is acceptable with respect to S in AF_2.*
4. *S is admissible in AF_2.*

Proof.

1. Since S defeats A in AF_1, there exist an argument B in S such that B defeats A in AF_1. According to the Property 3, there exist subarguments B' of B and A' of A such that B' attacks A' and B' defeats A in AF_1. Since AF_1 and AF_2 are right-equivalent, B' defeats A in AF_2. According to the Proposition 1, S is closed. Then B' belongs to S. Thus S defeats A in AF_2.
2. Since S is a complete extension of AF_1, it is conflict-free in AF_1. Let B and A be arguments in S such that B defeats A in AF_2. According to the Property 3, there exist subarguments B' of B and A' of A such that B' attacks A' and B' defeats A in AF_2. Since AF_1 and AF_2 are right-equivalent, B' defeats A in AF_1. According to the Proposition 1, S is closed. Then B' belongs to S. We have a contradiction since S is conflict-free in AF_1.
3. Let B be an argument such that B defeats A in AF_2. According to the Property 3, there exist subarguments B' of B and A' of A such that B' attacks A' and B' defeats A in AF_2. Since AF_1 and AF_2 are right-equivalent, B' defeats A in AF_1. Since A is acceptable with respect to S in AF_1, S defeats B' in AF_1. Then S defeats B' in AF_2. Thus A is acceptable with respect to S in AF_2.
4. Since S is a complete extension of AF_1, S is admissible in AF_1. Then S is conflict-free in AF_1 each argument in S is acceptable with respect to S in AF_1. Then S is conflict-free in AF_2 and each argument in S is acceptable with respect to S in AF_2. Thus S is admissible in AF_2. □

Proposition 2. *Let $(\mathcal{A}, \sqsubseteq, \mathcal{R})$ be a structured argumentation framework, $AF_1 = (\mathcal{A}, \mathcal{D}_1)$ and $AF_2 = (\mathcal{A}, \mathcal{D}_2)$ be abstract argumentation frameworks. If AF_1 and AF_2 are right-equivalent and satisfy the Properties 1 and 3 then complete (resp. preferred, stable, or grounded) extensions of AF_1 and AF_2 coincide.*

Proof. Let S be a complete extension of AF_1. Then S is admissible in AF_1 and S contains all arguments acceptable with respect to S in AF_1. According to the lemma 2, S is admissible in AF_2 and S contains all arguments acceptable with respect to S in AF_2. Thus S is a complete extension of AF_2. Similarly the opposite direction.

Preferred extension is a maximal (with respect to set inclusion) complete extension. Since complete extensions of AF_1 and AF_2 coincide, so do preferred extensions.

Stable extension is a complete extension S which defeat each argument A which does not belong to S. According to the lemma 2, S defeats A in AF_1 iff S defeats A in AF_2. Therefore stable extensions of AF_1 and AF_2 coincide.

Grounded extension is the least (with respect to set inclusion) complete extension. Since complete extensions of AF_1 and AF_2 coincide, so does grounded extension. □

Corollary 1. *Let $(\mathcal{A}, \sqsubseteq, \mathcal{R})$ be a structured argumentation framework, $AF_1 = (\mathcal{A}, \mathcal{D}_1)$ and $AF_2 = (\mathcal{A}, \mathcal{D}_2)$ be abstract argumentation frameworks. If AF_1 and AF_2 are right-equivalent, AF_1 satisfies the properties 2 and 3, and AF_2 satisfies the properties 1 and 4 then complete (resp. preferred, stable, or grounded) extensions of AF_1 and AF_2 coincide.*

Proof. Since the property 2 implies the property 1 and the property 4 implies the property 3, complete (resp. preferred, stable, or grounded) extensions of AF_1 and AF_2 coincide according to the previous proposition. □

We do not require that all attacks are successful. Similarly as in the case of propagation of defeat relation, the only reason for an unsuccessful attack is backward defeat.

Property 5. *If an argument A attacks an argument B then A defeats B or B defeats A.*

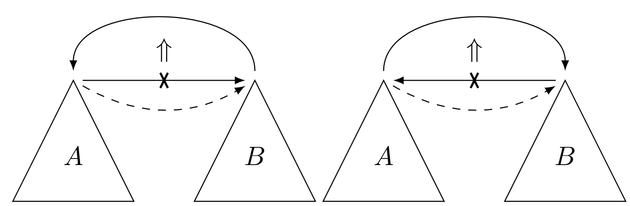

Figure 7: Consequences of the Property 5

Proposition 3. *Let $(\mathcal{A}, \sqsubseteq, \mathcal{R})$ be a structured argumentation framework and $AF = (\mathcal{A}, \mathcal{D})$ be an abstract argumentation framework. If AF satisfies the Property 5 then each complete extension of AF is consistent.*

Proof. Let S be a complete extension. Then S is conflict-free. Let A and B be arguments in S such that A attacks B. Then A defeats B or B defeats A. We have a contradiction since S is conflict-free. □

Instantiations

There exist many instantiations of abstract argumentation frameworks. We focus on two of them (Amgoud et al. 2005; Prakken and Sartor 1997), because they are well-known and both use some of the Dung's semantics. Since our analysis is based upon Dung's semantics (Dung 1995), approach by García and Simari (García and Simari 2004) cannot be directly investigated in this paper.

ASPIC

In (Amgoud et al. 2005), the underlying logical language is unspecified, but it is closed under negation.

Definition 4 (Defeasible Theory)**.** Let $\mathcal{L}$ be a language. A *strict rule* is an expression

$$\phi_1, \dots, \phi_n \rightarrow \phi_0$$

where $0 \leq n$ and each ϕ_i, $0 \leq i \leq n$, belongs to $\mathcal{L}$. A *defeasible rule* is an expression

$$\phi_1, \dots, \phi_n \Rightarrow \phi_0$$

where $0 \leq n$ and each ϕ_i, $0 \leq i \leq n$, belongs to $\mathcal{L}$. A *defeasible theory* is a pair $(\mathcal{S}, \mathcal{D})$ where $\mathcal{S}$ is a set of strict rules and $\mathcal{D}$ is a set of defeasible rules.

A transposition of a strict rule $\phi_1, \dots, \phi_n \rightarrow \phi_0$ is a strict rule of the form $\phi_1, \dots, \phi_{i-1}, \neg\phi_0, \phi_{i+1}, \dots, \phi_n \rightarrow \neg\phi_i$ for some $1 \leq i \leq n$.

Definition 5 (Closure)**.** Let $\mathcal{S}$ be a set of strict rules. The *closure of $\mathcal{S}$ under transposition* is a minimal set of strict rules $Cl_{tp}(\mathcal{S})$ satisfying

- $\mathcal{S} \subseteq Cl_{tp}(\mathcal{S})$
- if $r \in Cl_{tp}(\mathcal{S})$ and r' is a transposition of r then $r' \in Cl_{tp}(\mathcal{S})$

In the following, $\lceil . \rceil$ is objectification function introduced by Pollock (Prakken and Vreeswijk 2002) that transforms meta-language expressions into object-language.

Definition 6 (Argument)**.** Let $(\mathcal{S}, \mathcal{D})$ be a defeasible theory such that $Cl_{tp}(\mathcal{S}) = \mathcal{S}$. An *argument* is an expression A of the form

- $[A_1, \dots, A_n \rightarrow \psi]$ if $0 \leq n$ and $A_1, \dots, A_n$ are arguments such that $\textsc{Conc}(A_1), \dots, \textsc{Conc}(A_n) \rightarrow \psi$ belongs to $\mathcal{S}$

$$\begin{aligned}\textsc{Conc}(A) &= \psi\\ \textsc{Sub}(A) &= \textsc{Sub}(A_1) \cup \ldots \cup \textsc{Sub}(A_n) \cup \{A\}\\ \textsc{Top}(A) &= \textsc{Conc}(A_1), \dots, \textsc{Conc}(A_n) \rightarrow \psi\end{aligned}$$

- $[A_1, \dots, A_n \Rightarrow \psi]$ if $0 \leq n$ and $A_1, \dots, A_n$ are arguments such that $\textsc{Conc}(A_1), \dots, \textsc{Conc}(A_n) \Rightarrow \psi$ belongs to $\mathcal{D}$

$$\begin{aligned}\textsc{Conc}(A) &= \psi\\ \textsc{Sub}(A) &= \textsc{Sub}(A_1) \cup \ldots \cup \textsc{Sub}(A_n) \cup \{A\}\\ \textsc{Top}(A) &= \textsc{Conc}(A_1), \dots, \textsc{Conc}(A_n) \Rightarrow \psi\end{aligned}$$

An argument A is a subargument of an argument B if $A \in \textsc{Sub}(B)$. An argument A *undercuts* an argument B if $\textsc{Conc}(A) = \neg\lceil \textsc{Top}(B) \rceil$ and $\textsc{Top}(B) \in \mathcal{D}$. An argument A *rebuts* an argument B if $\textsc{Conc}(A) = \neg\,\textsc{Conc}(B)$ or $\textsc{Conc}(B) = \neg\,\textsc{Conc}(A)$. An argument A *restrictively rebuts* an argument B if A rebuts B and $\textsc{Top}(B) \in \mathcal{D}$.

Definition 7 (Attack)**.** An argument A *attacks* an argument B if A undercuts B or A restrictively rebuts B and $A \nprec B$.

Definition 8 (Defeat)**.** An argument A *defeats* an argument B if there exist subarguments A' of A and B' of B such that A' attacks B'.

In ASPIC, each attack is successful. Defeats always propagate upwards on the both sides.

Example 4. Tweety is a penguin. All penguins are birds. Birds usually fly. Penguins usually do not fly. It is not true that penguins usually fly because they are birds.

- $\phi_1\!: \rightarrow penguin(tweety)$
- $\phi_2\!: penguin(X) \rightarrow bird(X)$
- $\phi_3\!: bird(X) \Rightarrow flies(X)$
- $\phi_4\!: penguin(X) \Rightarrow \neg\, flies(X)$
- $\phi_5\!: penguin(X) \Rightarrow \neg\lceil bird(X) \Rightarrow flies(X) \rceil$

Five arguments can be constructed

- $A_1 = [\rightarrow penguin(tweety)]$
- $A_2 = [A_1 \rightarrow bird(tweety)]$
- $A_3 = [A_2 \Rightarrow flies(tweety)]$
- $A_4 = [A_1 \Rightarrow \neg\, flies(tweety)]$
- $A_5 = [A_1 \Rightarrow \neg\lceil bird(tweety) \Rightarrow flies(tweety) \rceil]$

We can see that the argument A_5 undercuts the argument A_3 and thus A_5 also defeats A_3. Similarly, the argument A_4 rebuts the argument A_3. If we prefer the more specific argument A_4 over the more general argument A_3, A_4 also defeats A_3.

Proposition 4. *A structured argumentation framework instantiated according to (Amgoud et al. 2005) satisfies the Property 2.*

Proof. Let A' be a subargument of A, B' be a subargument of B, and A' defeats B'. According to the Definition 8, there exist subarguments A'' of A' and B'' of B' such that A' attacks B'. Then A defeats B because A'' is a subargument of A and B'' is a subargument of B. □

Proposition 5. *A structured argumentation framework instantiated according to (Amgoud et al. 2005) satisfies the Property 3.*

Proof. Let argument A defeats an argument B. According to the Definition 8, there exist subarguments A' of A and B' of B such that A' attacks B'. Then A' also defeats B'. □

However a structured argumentation framework instantiated according to (Amgoud et al. 2005) does not satisfy the Property 4.

Example 5. Following arguments can be constructed from defeasible theory $(\emptyset, \{\Rightarrow p. \Rightarrow \neg\, p. p \Rightarrow a. \neg\, p \Rightarrow b.\})$

- $A_1 = [\Rightarrow p]$
- $A_2 = [\Rightarrow \neg p]$
- $A_3 = [p \Rightarrow a]$
- $A_4 = [\neg p \Rightarrow b]$

Argument A_3 defeats A_4, but there is no subargument B of A_4, such that A_3 attacks B.

Proposition 6. *A structured argumentation framework instantiated according to (Amgoud et al. 2005) satisfies the Property 5.*

Proof. Let argument A attacks an argument B. According to the Definition 8, A also defeats B. □

Prakken and Sartor

In (Prakken and Sartor 1997), argumentation system is based on the language of logic programs. Arguments are sequences of rules. They are not minimal in the sense that one argument may have many conclusions.

An *atom* is a propositional symbol. An *classical literal* is an atom or an atom preceded by classical negation $\neg$. A set of classical literals S is *incoherent* if S contains an atom A and a classical literal $\neg A$. An *default literal* is a classical literal preceded by default negation $\sim$. A set of classical literals S is *inconsistent* if S contains a classical literal L and a default literal $\sim L$.

Definition 9 (Defeasible Logic Program)**.** A *strict rule* is an expression of the form

$$L_1, \ldots, L_n \rightarrow L_0$$

where $0 \leq n$ and each L_i, $0 \leq i \leq n$, is a classical literal. We will denote

$$\begin{aligned} head(r) &= L_0 \\ body^+(r) &= \{L_1, \ldots, L_n\} \\ body^-(r) &= \emptyset \\ body(r) &= body^+(r) \cup body^-(r) \end{aligned}$$

A *defeasible rule* is an expression of the form

$$L_1, \ldots, L_m, \sim L_{m+1}, \ldots, \sim L_n \Rightarrow L_0$$

where $0 \leq m \leq n$ and each L_i, $0 \leq i \leq n$, is a classical literal. We will denote

$$\begin{aligned} head(r) &= L_0 \\ body^+(r) &= \{L_1, \ldots, L_m\} \\ body^-(r) &= \{\sim L_{m+1}, \ldots, \sim L_n\} \\ body(r) &= body^+(r) \cup body^-(r) \end{aligned}$$

A *defeasible logic program* is a pair $(\mathcal{S}, \mathcal{D})$ where $\mathcal{S}$ is a set of strict rules and $\mathcal{D}$ is a set of defeasible rules.

Definition 10 (Closure)**.** Let S be a set of classical literals and $\mathcal{S}$ be a set of strict rules. The *closure of* S *under* $\mathcal{S}$ is a minimal set of literals $Cl_{\mathcal{S}}(S)$ satisfying

- $S \subseteq Cl_{\mathcal{S}}(S)$
- if $body(r) \subseteq Cl_{\mathcal{S}}(S)$ then $head(r) \in Cl_{\mathcal{S}}(S)$ for each $r \in \mathcal{S}$

Definition 11 (Argument)**.** Let $(\mathcal{S}, \mathcal{D})$ be a defeasible logic program. An *argument* is a sequence of rules $A = [r_0, \ldots, r_n]$, $0 \leq n$, such that for each $0 \leq i \leq n$

- $body^+(r_i) \subseteq \{head(r_j) \mid 0 \leq j < i\}$
- $head(r_i) \notin \{head(r_j) \mid 0 \leq j < i\}$

We will denote

$$\begin{aligned} \textsc{Concs}(A) &= Cl_{\mathcal{S}}(\{head(r_i) \mid 0 \leq i \leq n\}) \\ \textsc{Ass}(A) &= \bigcup_{0 \leq i \leq n} body^-(r_i) \end{aligned}$$

An argument A is a *subargument of* an argument B if A is a subsequence of B. An argument A *rebuts* an argument B if $\neg\,\textsc{Concs}(A) \cap \textsc{Concs}(B) \neq \emptyset$. An argument A *undercuts* an argument B if $\sim \textsc{Concs}(A) \cap \textsc{Ass}(B) \neq \emptyset$. An argument A is *incoherent* if A rebuts A or A undercuts A.

Definition 12 (Attack)**.** An argument A *attacks* an argument B if

- A undercuts B, or
- A rebuts B and $A \not\prec B$, or
- B is incoherent.

Definition 13 (Defeat)**.** An argument A *defeats* an argument B if

- A undercuts B, or
- A rebuts B, $A \not\prec B$, and B does not undercut A, or
- B is incoherent.

Example 6. Let P be the following defeasible logic program

$$\begin{array}{llll} r_1: & \sim a & \Rightarrow & \neg b \\ r_2: & & \Rightarrow & b \\ r_3: & b & \Rightarrow & a \end{array}$$

and $r_2 \prec r_1$. Consider these three arguments $A_1 = [r_1]$, $A_2 = [r_2]$, $A_3 = [r_2, r_3]$, where A_2 is a subargument of A_3. A_1 attacks both A_2 and A_3 and A_3 attacks A_1. Also A_1 defeats A_2 and A_3 defeats A_1, but A_1 does not defeat A_3 since it is undercutted.

Example 7. Let P be the following defeasible logic program

$$\begin{array}{llll} r_1: & & \Rightarrow & a \\ r_2: & \sim b, a & \Rightarrow & c \\ r_3: & & \Rightarrow & \neg a \\ r_4: & \neg a & \Rightarrow & b \end{array}$$

Consider following arguments $A_1 = [r_1]$, $A_2 = [r_1, r_2]$, $A_3 = [r_3, r_4]$. We can see that A_1 defeats A_3 with rebutting, but A_2 does not, because it is undercutted by A_3.

The previous examples showed us that in formalism (Prakken and Sartor 1997) defeat relation does not propagate upwards through superarguments on the right or left side: if an argument A defeats an argument B then neither A defeats all superarguments of B (Example 6), nor all superarguments of A defeats B (Example 7). However (Prakken and Sartor 1997) still satisfies the Property 2.

Proposition 7. *A structured argumentation framework instantiated according to (Prakken and Sartor 1997) satisfies the Property 2.*

Proof. Let A' be a subargument of an argument A and B' be a subargument of an argument B such that A' defeats B'. According to the Definition 13, A' undercuts B', or A' rebuts B', $A' \not\prec B'$, and B' does not undercut A', or B' is incoherent. Let A' undercut B'. Then A undercuts B and thus A defeats B. Let A' rebut B', $A' \not\prec B'$, and B' do not undercut A'. Then A rebuts B and B rebuts A. If B undercuts A then B defeats A. If A undercuts B then A defeats B. Let B does not undercut A and A does not undercut B. If $A \prec B$ then B defeats A. If $A \not\prec B$ then A defeats B. Let B' be incoherent. Then B is incoherent and thus A defeats B. □

Proposition 8. *A structured argumentation framework instantiated according to (Prakken and Sartor 1997) satisfies the Property 3.*

Proof. Let an argument A defeat an argument B. According to the Definition 13, A undercuts B, or A rebuts B, $A \not\prec B$, and B does not undercut A, or B is incoherent. According to the Definition 12, A attacks B in all cases. □

Proposition 9. *A structured argumentation framework instantiated according to (Prakken and Sartor 1997) satisfies the Property 5.*

Proof. Let an argument A attacks an argument B. According to the Definition 12, A undercuts B, or A rebuts B and $A \not\prec B$, or B is incoherent. If B undercuts A then B defeats A. Let B do not undercut A. According to the Definition 13, A defeats B in all cases. □

Conclusion

In this paper we introduced a structured argumentation framework, where only a set of abstract arguments partially ordered by a subargument relation, and an attack relation among arguments is considered. We have studied on the abstract level consequences of some design decisions how to define defeat relation in terms of structured arguments and attack relation.

Compositionality principle (Prakken and Vreeswijk 2002) saying that an argument cannot be justified unless all its subarguments are justified, is one of the most accepted one and also satisfied by existing well known approaches (Prakken and Sartor 1997; Amgoud et al. 2005; García and Simari 2004). We have shown that compositionality principle can be guaranteed under all traditional Dung's semantics, no matter what are the definitions of argument and attack, it is sufficient that defeat relation satisfies some abstract property.

We have also shown sufficient condition for consistency with respect to attack relation. We have seen that propagation of the defeat relation on the left side does not affect standard Dung's semantics.

We believe that there are two different viewpoints on formal systems: the first one is interesting for theoretical analysis, where ones prefers definitions as much general as possible. The second one is an implementation viewpoint, where a programmer prefers the most specific definitions, which are usually easier to implement and may be more efficient. Thus, this paper successfully confirmed the relevance of studying various design decisions of formal argumentation systems within the structured argumentation framework.

References

Amgoud, L.; Caminada, M.; Doutre, S.; Prakken, H.; and Vreeswijk, G. A. W. 2005. Draft formal semantics for ASPIC system. Technical report.

Caminada, M., and Amgoud, L. 2007. On the evaluation of argumentation formalisms. *Artificial Intelligence* 171(5-6):286–310.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77(2):321–357.

García, A. J., and Simari, G. R. 2004. Defeasible Logic Programming: An Argumentative Approach. *Theory and Practice of Logic Programming* 4(1-2):95–138.

Martínez, D. C.; García, A. J.; and Simari, G. R. 2006. On Acceptability in Abstract Argumentation Frameworks with an Extended Defeat Relation. In *Computational Models of Argument*, volume 144 of *Frontiers in Artificial Intelligence and Applications*, 273–278. IOS Press.

Prakken, H., and Sartor, G. 1997. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Nonclassical Logics* 7(1):25–75.

Prakken, H., and Vreeswijk, G. 2002. *Handbook of Philosophical Logic*, volume 4. Kluwer Academic Publishers. chapter 3, 219–318.

Prakken, H. 2010. An abstract framework for argumentation with structured arguments. *Argument & Computation* 1(2):93–124.