Computing Preferred Extensions for Argumentation Systems with Sets of Attacking Arguments

Søren Holbech Nielsen [a], Simon Parsons [b]
[a] *Department of Computer Science*
Aalborg University, Aalborg
Denmark
[b] *Department of Computer and Information Science*
Brooklyn College, City University of New York
Brooklyn, 11210 NY, USA

AbstractThe hitherto most abstract, and hence general, argumentation system, is the one described by Dung in a paper from 1995. This framework does not allow for joint attacks on arguments, but in a recent paper we adapted it to support such attacks, and proved that this adapted framework enjoyed the same formal properties as that of Dung. One problem posed by Dung's original framework, which was neglected for some time, is how to compute preferred extensions of the argumentation systems. However, in 2001, in a paper by Doutre and Mengin, a procedure was given for enumerating preferred extensions for these systems. In this paper we propose a method for enumerating preferred extensions of the potentially more complex systems, where joint attacks are allowed. The method is inspired by the one given by Doutre and Mengin.

Keywords. Argumentation with sets, Preferred Extensions,

1. Introduction

In the last fifteen years or so, there has been much interest in argumentation systems within the artificial intelligence community. This interest spreads across many different sub-areas of artificial intelligence. One of these is non-monotonic reasoning [1,2], which exploits the fact that argumentation systems can handle, and resolve, inconsistencies [3, 4] and uses it to develop general descriptions of non-monotonic reasoning [5,6]. This line of work is summarised in [7]. Another area that makes use of argumentation is reasoning and decision making under uncertainty [8,9,10], which exploits the dependency structure one can infer from arguments in order to correctly combine evidence. Much of this work is covered in [11]. More recently [12,13], the multi-agent systems community has begun to make use of argumentation, using it to develop a notion of rational interaction [14,15].

One very influential and very abstract system of argumentation was that introduced by Dung [16]. This was, for instance, the basis for the work in [5], was the system extended by Amgoud in [17,18], and subsequently as the basis for the dialogue systems in [19,20]. The importance of Dung's results is mainly due to the fact that his framework abstracts away from details of language and argumentation rules, that the presented semantics therefore are clear and intuitive, and that relationships among arguments can be analysed in isolation from other (e.g. implicational) relationships. Furthermore, the results can easily be transferred to any other argumentation framework, by identifying that framework's equivalent of an attack. It is this generality, we believe, that has contributed to the popularity of the work, and we see it as a prime contender for becoming an established standard for further investigations into the nature of arguments and their interaction.

However, even if Dung was trying to abstract away from the underlying language and structure of arguments, his framework implicitly assumes a logical "and" connective in the underlying language, to be able to model all kinds of attacks. This hidden assumption is caused by Dung's attack relation being a simple binary relation from one argument to another, rather than a relation mapping sets of arguments to other sets of arguments. In a recent paper [21] we presented a generalisation of Dung's framework, which allows sets of arguments to attack single arguments, and thus frees the underlying language from being closed under some logical "and" connective[1]. The main motivation for that work was that sometimes it seems reasonable for a number of arguments to interact and constitute an attack on some other argument, even though the arguments of the attack does not individually attack that argument. The approach, where such joint attacks are modelled by adding to the argumentation system a new argument that represents the set of attacking arguments, and then employing traditional argumentation analysis to this system, is not satisfactory: The encoding is artificial, adding distance between the formalism and the modelled argumentation situation, and to ensure that nonsense conclusions do not arise, the relation of attack among arguments need to be restricted or an extra layer of logical relationships among arguments need to be specified. The former muddles the clear distinction between arguments and attacks, which was the very appeal of Dung's framework, and the latter makes it hard to survey the effects of one set of argument on others and calls for more specialized formalisms for analysis than Dung's. For further elaborations on this see [21].

In this paper, we build on the work in [21] and propose a method for enumerating preferred extensions of the argumentation systems defined there. In general it is hard to compute a preferred extension [23], but [24] presents a method that enumerates preferred extensions for an abstract argumentation system as presented in [16]. Moreover, [25] and [26] present methods for answering whether a specific argument is in at least one preferred extension, or if it is in all preferred extensions. Here we adapt the basic

[1] Subsequently, we have been directed to [22], which describes an argumentation framework that is a generalization of that in [16] too. The main differences between [22] and [21] are due to difference in perspectives: Bochman is motivated by the task of establishing a correspondence between disjunctive logic programming and abstract argumentation, and ends up with a framework that allows any finite set of arguments (including the empty set) to attack and be attacked by any other finite set, whereas we have tried to expand the dialogical and dialectical boundaries of abstract argumentation by allowing for arbitrary sets of attacking arguments (but the empty set), and claim that further flexibility is not needed for argumentative reasoning. (Indeed, the main example motivating attacks on entire sets of arguments turns out to be sensibly represented in our framework.) Due to his aims, Bochman construct new semantics for his framework and identifies new families of argumentation systems with nice properties (none of them coinciding with our formalism). We, on the other hand, stick as close as possible to the semantics provided by Dung, and instead show that the all of Dung's results are valid for systems with sets of attacking arguments.

technique of [24] to the more complex case of argumentation systems with joint attacks. The main problem for this adaptation, is that the argumentation systems of [16] can be viewed as directed graphs, and that this fact is exploited in the pruning rules of [24]. For the argumentation systems of [21], however, no similar graph structure exists, and new pruning rules thus have to be constructed. In particular, we lack a context independent notion of "reflective" arguments, and a context independent notion of a single argument being detrimental to a specific set of arguments.

2. Argumentation With Attacking Sets of Arguments

In this section we present our generalisation of the framework of [16], as introduced in [21].

Definition 1 (Argumentation Systems)**.** *An* argumentation system *is a pair* $(\boldsymbol{A}, \triangleright)$, *where* $\boldsymbol{A}$ *is a set of arguments, and* $\triangleright \subseteq (\mathcal{P}(\boldsymbol{A}) \setminus \{\varnothing\}) \times \boldsymbol{A}$ *is an* attack relation.

Throughout the paper we assume an argumentation system $\mathcal{A} = (\boldsymbol{A}, \triangleright)$, and take it to be implicit.

We say that a set of arguments $\boldsymbol{S}$ *attacks* an argument A, if there is $\boldsymbol{S}' \subseteq \boldsymbol{S}$ such that $\boldsymbol{S}' \triangleright A$. In that case we also say that A *is attacked by* $\boldsymbol{S}$. If there is no set $\boldsymbol{S}'' \subsetneq \boldsymbol{S}'$ such that $\boldsymbol{S}''$ attacks A, then we say that $\boldsymbol{S}'$ is a *minimal* attack on A. Obviously, if there is a set that attacks an argument A, then there must also exist a minimal attack on A. Moreover, if $\boldsymbol{S}$ is a minimal attack on A, then it must be the case that $\boldsymbol{S} \triangleright A$. If for two sets of arguments $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, there is an argument $A \in \boldsymbol{S}_2$, which is attacked by $\boldsymbol{S}_1$, then we say that $\boldsymbol{S}_1$ attacks $\boldsymbol{S}_2$, and that $\boldsymbol{S}_2$ is attacked by $\boldsymbol{S}_1$. If a set $\boldsymbol{S}_1$ attacks some argument in $\boldsymbol{S}_2$, and this is true of no subsets of $\boldsymbol{S}_1$, then we say that $\boldsymbol{S}_1$ is a minimal attack on $\boldsymbol{S}_2$, and relaxing notation a bit, write $\boldsymbol{S}_1 \triangleright \boldsymbol{S}_2$. If a set of arguments $\boldsymbol{S}$ does not attack itself, then we say that $\boldsymbol{S}$ is *conflict-free*.

Let $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ be sets of arguments. If $\boldsymbol{S}_2$ attacks some argument A, and $\boldsymbol{S}_1$ attacks $\boldsymbol{S}_2$, then we say that $\boldsymbol{S}_1$ is a *defense of* A *from* $\boldsymbol{S}_2$, and that $\boldsymbol{S}_1$ *defends* A *from* $\boldsymbol{S}_2$. Obviously, if $\boldsymbol{S}_3$ is a superset of $\boldsymbol{S}_1$, $\boldsymbol{S}_3$ is also a defense of A from $\boldsymbol{S}_2$. An argument A is said to be *acceptable with respect to a set of arguments* $\boldsymbol{S}$, if $\boldsymbol{S}$ defends A from all sets of attacking arguments $\boldsymbol{S}' \subseteq \boldsymbol{A}$. A conflict-free set of arguments $\boldsymbol{S}$ is said to be *admissible* if each argument in $\boldsymbol{S}$ is acceptable with respect to $\boldsymbol{S}$. This leads us to the credulous semantics we treat in this paper:

Definition 2 (Preferred Extensions)**.** *An admissible set* $\boldsymbol{S}^*$ *is called a* preferred extension*, if there is no admissible set* $\boldsymbol{S}' \subseteq \boldsymbol{A}$*, such that* $\boldsymbol{S}^* \subsetneq \boldsymbol{S}'$.

From [16] and [21], we have that for each admissible set $\boldsymbol{S}$, there exists a preferred extension $\boldsymbol{S}^*$, such that $\boldsymbol{S} \subseteq \boldsymbol{S}^*$. Moreover, as the empty set is an admissible set, we have that every argumentation system has at least one preferred extension.

A very skeptical semantics, is the *grounded extension*, which is defined as the least fix point of the function $F : \mathcal{P}(\boldsymbol{A}) \rightarrow \mathcal{P}(\boldsymbol{A})$, defined as

$$F(\boldsymbol{S}) = \{A \ : \ A \text{ is acceptable wrt. } \boldsymbol{S}\}.$$

Example 1 (An Introductory Example): *Consider an argumentation system* $\mathcal{A}_e = (\boldsymbol{A}_e, \triangleright_e)$*, where* $\boldsymbol{A}_e = \{A, B, C, D, E, F\}$ *and* $\triangleright_e$ *is defined as:*

$$\{A, C, D\} \triangleright_e B, \quad \{A, B\} \triangleright_e C, \quad \{B\} \triangleright_e D, \quad \{C, E\} \triangleright_e D,$$
$$\{D\} \triangleright_e E, \quad \{B, F\} \triangleright_e E, \quad \{A\} \triangleright_e F, \text{ and } \{D\} \triangleright_e F.$$

It can easily be verified that the grounded extension of $\mathcal{A}_e$ *is* $\{A\}$*. The preferred extensions are* $\{A, B, E\}$ *and* $\{A, C, D\}$*, which we shall prove later in the paper.*

3. Computing Preferred Extensions

We now present a method for computing the preferred extensions for an argumentation system with sets of attacking arguments as defined in Definition 1. The method is inspired by a similar method, for computing preferred extensions for Dung's original argumentation systems, presented in [24]. The basic strategy is to enumerate all possible divisions of $\boldsymbol{A}$ into two sets, $\boldsymbol{I}$ and $\boldsymbol{O}$, where $\boldsymbol{I}$ are the arguments that are *i*n a preferred extension, and $\boldsymbol{O}$ are those that are *o*ut, and then check for each division if $\boldsymbol{I}$ is a preferred extension. Now, of course the number of divisions can be drastically reduced, by noting requirements on $\boldsymbol{I}$ imposed by Definition 2, so a full enumeration can often be avoided.

The enumeration of divisions is constructed as a tree, where each node is a partition of $\boldsymbol{A}$ into three sets $(\boldsymbol{I}, \boldsymbol{O}, \boldsymbol{U})$, where $\boldsymbol{U}$ is the arguments still not assigned to one of the two divisions $\boldsymbol{I}$ and $\boldsymbol{O}$. The root of the tree is a node where both $\boldsymbol{I}$ and $\boldsymbol{O}$ are empty and all arguments are assigned to the undecided partition. Each child $(\boldsymbol{I}', \boldsymbol{O}', \boldsymbol{U}')$ of a node $(\boldsymbol{I}, \boldsymbol{O}, \boldsymbol{U})$ is then a refinement of the division represented by the previous node, i.e. $\boldsymbol{I} \subseteq \boldsymbol{I}'$ and $\boldsymbol{O} \subseteq \boldsymbol{O}'$. The size of such a tree is exponential in the number of arguments, but fortunately we often do not have to construct the entire tree, and if only more specific queries are sought answered (such as "Is argument A included in some preferred extension?") we can sometimes get away with only inspecting parts of a few branches of the tree.

First we define the nodes we work with. These are called $\mathcal{A}$-*candidates*, or as we take $\mathcal{A}$ to be implicit, just *candidates*. For a given set $\boldsymbol{S} \subseteq \boldsymbol{A}$, define

$$\boldsymbol{S}^{\rightarrow} = \{A \in \boldsymbol{A} \ : \ \exists \boldsymbol{T} \subseteq \boldsymbol{S} \text{ s.t. } \boldsymbol{T} \triangleright A\}$$

and

$$\boldsymbol{S}^{\leftarrow} = \{A \in \boldsymbol{A} \ : \ \exists \boldsymbol{T} \subseteq \boldsymbol{S}, B \in \boldsymbol{S} \text{ s.t. } \boldsymbol{T} \cup \{A\} \triangleright B\}.$$

$\boldsymbol{S}^{\rightarrow}$ is thus the set of arguments attacked by $\boldsymbol{S}$, and $\boldsymbol{S}^{\leftarrow}$ is the set of arguments, which if added to $\boldsymbol{S}$, would make $\boldsymbol{S}$ attack itself. A candidate is then a triple $(\boldsymbol{I} \subseteq \boldsymbol{A}, \boldsymbol{O} \subseteq \boldsymbol{A}, \boldsymbol{U} = \boldsymbol{A} \setminus (\boldsymbol{I} \cup \boldsymbol{O}))$ satisfying the following properties:

$$\boldsymbol{I}^{\rightarrow} \subseteq \boldsymbol{O}, \tag{1}$$
$$\boldsymbol{I}^{\leftarrow} \subseteq \boldsymbol{O}, \text{ and} \tag{2}$$
$$\boldsymbol{I} \cap \boldsymbol{O} = \varnothing. \tag{3}$$

(If $\mathcal{C} = (\boldsymbol{I}, \boldsymbol{O}, \boldsymbol{U})$ is a triple, we will use subscripts to refer to the sets in the partition, e.g. $\boldsymbol{I}_\mathcal{C}$ denotes the set $\boldsymbol{I}$ in $\mathcal{C}$.)

Example 2 (Candidates): *We consider again the argumentation system $\mathcal{A}_e = (\boldsymbol{A}_e, \triangleright_e)$ from Example 1. A few examples of candidates are $(\{B\}, \{D\}, \{A, C, E, F\})$, $(\varnothing, \boldsymbol{A}_e, \varnothing)$, and $(\{C, D\}, \{A, B, E, F\}, \varnothing)$. Some examples of non-candidates are $(\{A\}, \{B, D\}, \{C, E, F\})$, $(\{E, F\}, \{A, D\}, \{B, C\})$, and $(\{B\}, \{B, D\}, \{A, C, E, F\})$.*

Focusing only on candidates, rather than arbitrary divisions of $\boldsymbol{A}$, is thus a restriction on the number of divisions to consider. We argue that it is sufficient below.

It follows from (1) and (3), that for any candidate $\mathcal{C}$, $\boldsymbol{I}_\mathcal{C}$ is conflict-free. For any triple $\mathcal{C}$, we denote by pref($\mathcal{C}$) the set of all preferred extensions $\boldsymbol{S}^*$, where $\boldsymbol{I}_\mathcal{C} \subseteq \boldsymbol{S}^* \subseteq \boldsymbol{I}_\mathcal{C} \cup \boldsymbol{U}_\mathcal{C}$. It follows, that if $\boldsymbol{U}_\mathcal{C} = \varnothing$, then pref($\mathcal{C}$) is $\{\boldsymbol{I}_\mathcal{C}\}$ if $\boldsymbol{I}_\mathcal{C}$ is a preferred extension and $\varnothing$ otherwise.

Given a triple $\mathcal{C}$ and an argument $A \in \boldsymbol{U}_\mathcal{C}$, define the triples

$$\mathcal{C} - A = (\boldsymbol{I}_\mathcal{C}, \quad \boldsymbol{O}_\mathcal{C} \cup \{A\}, \quad \boldsymbol{U}_\mathcal{C} \setminus \{A\}), \tag{4}$$

and

$$\mathcal{C} + A = (\boldsymbol{I}_\mathcal{C} \cup \{A\}, \quad \boldsymbol{O}_\mathcal{C} \cup \boldsymbol{\Delta}^{\rightarrow}_{\mathcal{C}+A} \cup \boldsymbol{\Delta}^{\leftarrow}_{\mathcal{C}+A}, \quad \boldsymbol{U}_\mathcal{C} \setminus (\{A\} \cup \boldsymbol{\Delta}^{\rightarrow}_{\mathcal{C}+A} \cup \boldsymbol{\Delta}^{\leftarrow}_{\mathcal{C}+A})) \tag{5}$$

where

$$\boldsymbol{\Delta}^{\rightarrow}_{\mathcal{C}+A} = \{B \in \boldsymbol{U}_\mathcal{C} \ : \ \exists \boldsymbol{S} \subseteq \boldsymbol{I}_\mathcal{C} \text{ s.t. } \boldsymbol{S} \cup \{A\} \triangleright B\} \tag{6}$$

and

$$\begin{aligned} \boldsymbol{\Delta}^{\leftarrow}_{\mathcal{C}+A} = \{B \in \boldsymbol{U}_\mathcal{C} \ : \ & \exists \boldsymbol{S} \subseteq \boldsymbol{I}_\mathcal{C}, C \in \boldsymbol{I}_\mathcal{C} \text{ s.t. } \boldsymbol{S} \cup \{B\} \triangleright A \\ & \vee \boldsymbol{S} \cup \{A, B\} \triangleright C \ \vee \ \boldsymbol{S} \cup \{A, B\} \triangleright A\}. \end{aligned} \tag{7}$$

Example 3 (Adding Arguments to Triples): *Building on Example 2, we add E to the candidate $\mathcal{C}_1 = (\{B\}, \{D\}, \{A, C, E, F\})$ and the non-candidate $\mathcal{C}_2 = (\{A\}, \{B, D\}, \{C, E, F\})$: In the first case, $\boldsymbol{\Delta}^{\rightarrow}_{\mathcal{C}_1+E} = \varnothing$ and $\boldsymbol{\Delta}^{\leftarrow}_{\mathcal{C}_1+E} = \{F\}$, and in the second $\boldsymbol{\Delta}^{\rightarrow}_{\mathcal{C}_2+E} = \boldsymbol{\Delta}^{\leftarrow}_{\mathcal{C}_2+E} = \varnothing$. Therefore, $\mathcal{C}_1 + E = (\{B, E\}, \{D, F\}, \{A, C\})$ and $\mathcal{C}_2 + E = (\{A, E\}, \{B, D\}, \{C, F\})$.*

It is easy to verify that, given a candidate $\mathcal{C}$ and an argument $A \in \boldsymbol{U}_\mathcal{C}$, we have that

$$\boldsymbol{I}^{\rightarrow}_{\mathcal{C}+A} \setminus \boldsymbol{I}^{\rightarrow}_{\mathcal{C}} = \boldsymbol{\Delta}^{\rightarrow}_{\mathcal{C}+A}, \tag{8}$$

and

$$\boldsymbol{I}^{\leftarrow}_{\mathcal{C}+A} \setminus \boldsymbol{I}^{\leftarrow}_{\mathcal{C}} = \boldsymbol{\Delta}^{\leftarrow}_{\mathcal{C}+A}. \tag{9}$$

Given the partial division represented by a candidate, some arguments might be impossible to add to the set $\boldsymbol{I}$ without ending up with a contradiction. We therefore define the set of *reflexive arguments* with respect to a candidate $\mathcal{C}$ as follows:

$$\text{refl}(\mathcal{C}) = \{A \in \boldsymbol{U}_\mathcal{C} \ : \ \exists \boldsymbol{S} \subseteq \boldsymbol{I}_\mathcal{C}, \text{ s.t. } \boldsymbol{S} \cup \{A\} \triangleright A\}. \tag{10}$$

From the definitions, it immediately follows that if $A \in \text{refl}(\mathcal{C})$ then $\text{pref}(\mathcal{C} + A) = \varnothing$. Furthermore, we can state an important theorem, which implies that given a candidate $\mathcal{C}$, we can use the definitions of $\mathcal{C} + A$ and $\mathcal{C} - A$ to construct a tree of candidates having $\mathcal{C}$ as root:

Theorem 1. *Let $\mathcal{C}$ be a candidate, and $A \in \boldsymbol{U}_\mathcal{C}$. If $A \notin$ refl$(\mathcal{C})$ then both $\mathcal{C} + A$ and $\mathcal{C} - A$ are candidates as well. Otherwise only $\mathcal{C} - A$ is a candidate.*

Proof. It is obvious that $\mathcal{C} - A$ is a candidate no matter whether A is in $\text{refl}(\mathcal{C})$ or not. We therefore only show that $\mathcal{C} + A$ is a candidate iff A is not in $\text{refl}(\mathcal{C})$.

First, assume that A is in $\text{refl}(\mathcal{C})$. This means that there is some set $\boldsymbol{S} \subseteq \boldsymbol{I}_\mathcal{C}$, such that $\boldsymbol{S} \cup \{A\} \triangleright A$. Consequently, $\boldsymbol{I}_{\mathcal{C}+A} = \boldsymbol{I}_\mathcal{C} \cup \{A\}$ contains a subset $\boldsymbol{T} = \boldsymbol{S} \cup \{A\}$, such that $\boldsymbol{T} \triangleright A$. If $\mathcal{C} + A$ was to be a candidate, (1) would therefore require that A is in $\boldsymbol{O}_{\mathcal{C}+A}$. It follows that A is in $\boldsymbol{I}_{\mathcal{C}+A} \cap \boldsymbol{O}_{\mathcal{C}+A}$, which is thus not empty. That contradicts (3), and $\mathcal{C} + A$ can thus not be a candidate.

Conversely, assume that A is not in $\text{refl}(\mathcal{C})$, and we show that $\mathcal{C} + A$ is a candidate by means of contradiction. That is, assume that $\mathcal{C} + A$ is not a candidate, which means that one of the following must be true:

(i): $\exists B \in \boldsymbol{I}^{\rightarrow}_{\mathcal{C}+A}$ s.t. $B \notin \boldsymbol{O}_{\mathcal{C}+A}$,
(ii): $\exists B \in \boldsymbol{I}^{\leftarrow}_{\mathcal{C}+A}$ s.t. $B \notin \boldsymbol{O}_{\mathcal{C}+A}$, or
(iii): $\exists B \in \boldsymbol{I}_{\mathcal{C}+A} \cap \boldsymbol{O}_{\mathcal{C}+A}$.

We show that each case is impossible. First, assume that (i) is the case. Since $\mathcal{C}$ is a candidate, we necessarily have that $\boldsymbol{I}^{\rightarrow}_\mathcal{C} \subseteq \boldsymbol{O}_\mathcal{C} \subseteq \boldsymbol{O}_{\mathcal{C}+A}$ and it must thus be the case that $B \in \boldsymbol{I}^{\rightarrow}_{\mathcal{C}+A} \setminus \boldsymbol{I}^{\rightarrow}_\mathcal{C}$, which according to (8) is equivalent to having $B \in \boldsymbol{\Delta}^{\rightarrow}_{\mathcal{C}+A}$. But according to (5), $\boldsymbol{\Delta}^{\rightarrow}_{\mathcal{C}+A}$ is a subset of $\boldsymbol{O}_{\mathcal{C}+A}$, so $B \in \boldsymbol{O}_{\mathcal{C}+A}$ after all, which is a contradiction. Case (ii) is proved to be impossible with a similar argument.

Assume that (iii) is the case. Since $\mathcal{C}$ is a candidate, we know from (3) that $\boldsymbol{O}_\mathcal{C} \cap \boldsymbol{I}_\mathcal{C} = \varnothing$, and, since A is in $\boldsymbol{U}_\mathcal{C}$, which is disjoint from $\boldsymbol{O}_\mathcal{C}$, also that $\boldsymbol{O}_\mathcal{C} \cap (\boldsymbol{I}_\mathcal{C} \cup \{A\}) = \boldsymbol{O}_\mathcal{C} \cap \boldsymbol{I}_{\mathcal{C}+A} = \varnothing$. Therefore, B must be a member of $\boldsymbol{O}_{\mathcal{C}+A} \setminus \boldsymbol{O}_\mathcal{C} = (\boldsymbol{\Delta}^{\rightarrow}_{\mathcal{C}+A} \cup \boldsymbol{\Delta}^{\leftarrow}_{\mathcal{C}+A}) \subseteq \boldsymbol{U}_\mathcal{C}$. Furthermore, as $\boldsymbol{U}_\mathcal{C} \cap \boldsymbol{I}_\mathcal{C} = \varnothing$ it follows that B must be in $\boldsymbol{I}_{\mathcal{C}+A} \setminus \boldsymbol{I}_\mathcal{C} = \{A\}$. Thus, A must be in either $\boldsymbol{\Delta}^{\rightarrow}_{\mathcal{C}+A}$ or $\boldsymbol{\Delta}^{\leftarrow}_{\mathcal{C}+A}$. The first possibility is ruled out, since A by assumption is not a member of $\text{refl}(\mathcal{C})$. So A must be in $\boldsymbol{\Delta}^{\leftarrow}_{\mathcal{C}+A}$.

According to the definition of $\boldsymbol{\Delta}^{\leftarrow}_{\mathcal{C}+A}$, there must be a $C \in \boldsymbol{I}_\mathcal{C}$ and a set $\boldsymbol{S} \subseteq \boldsymbol{I}_\mathcal{C}$, so either $\boldsymbol{S} \cup \{A\} \triangleright A$ or $\boldsymbol{S} \cup \{A\} \triangleright C$. Again the first possibility is precluded by the assumption that A is not in $\text{refl}(\mathcal{C})$, so it must be the case that $\boldsymbol{S} \cup \{A\} \triangleright C$. But then A is in $\boldsymbol{I}^{\leftarrow}_\mathcal{C}$ and as $\mathcal{C}$ is a candidate also in $\boldsymbol{O}_\mathcal{C}$. That contradicts the assumption that A is in $\boldsymbol{U}_\mathcal{C}$, and the theorem follows. □

The theorem thus establishes that iterated use of the $\mathcal{C} + A$ and $\mathcal{C} - A$-definitions makes sense. Moreover, we have the following result on that activity:

Theorem 2. *Let $\mathcal{C}$ be a candidate and A and B be distinct arguments in $\boldsymbol{U} \setminus$ refl$(\mathcal{C})$, such that both $(\mathcal{C} + A) + B$ and $(\mathcal{C} + B) + A$ are candidates. Then*

$$(\mathcal{C}+A)+B=(\mathcal{C}+B)+A, \tag{11}$$

$$(\mathcal{C}-A)-B=(\mathcal{C}-B)-A, \textit{ and} \tag{12}$$

$$(\mathcal{C}+A)-B=(\mathcal{C}-B)+A. \tag{13}$$

Proof. We only show (11), since the others follow from similar, albeit slightly simpler arguments. It is obvious that $\boldsymbol{I}_{(\mathcal{C}+A)+B} = \boldsymbol{I}_{(\mathcal{C}+B)+A}$ and, given that $\boldsymbol{O}_{(\mathcal{C}+A)+B} = \boldsymbol{O}_{(\mathcal{C}+B)+A}$, also that $\boldsymbol{U}_{(\mathcal{C}+A)+B} = \boldsymbol{U}_{(\mathcal{C}+B)+A}$. We therefore just need to show that $\boldsymbol{O}_{(\mathcal{C}+A)+B} = \boldsymbol{O}_{(\mathcal{C}+B)+A}$:

$$\begin{aligned}\boldsymbol{O}_{(\mathcal{C}+A)+B} =&\boldsymbol{O}_{\mathcal{C}} \cup \boldsymbol{\Delta}^{\rightarrow}_{\mathcal{C}+A} \cup \boldsymbol{\Delta}^{\leftarrow}_{\mathcal{C}+A} \cup \boldsymbol{\Delta}^{\rightarrow}_{(\mathcal{C}+A)+B} \cup \boldsymbol{\Delta}^{\leftarrow}_{(\mathcal{C}+A)+B}\\
=&\boldsymbol{O}_{\mathcal{C}} \cup (\boldsymbol{I}^{\rightarrow}_{\mathcal{C}+A} \setminus \boldsymbol{I}^{\rightarrow}_{\mathcal{C}}) \cup (\boldsymbol{I}^{\leftarrow}_{\mathcal{C}+A} \setminus \boldsymbol{I}^{\leftarrow}_{\mathcal{C}})\\
&\cup (\boldsymbol{I}^{\rightarrow}_{(\mathcal{C}+A)+B} \setminus \boldsymbol{I}^{\rightarrow}_{\mathcal{C}+A}) \cup (\boldsymbol{I}^{\leftarrow}_{(\mathcal{C}+A)+B} \setminus \boldsymbol{I}^{\leftarrow}_{\mathcal{C}+A})\\
=&\boldsymbol{O}_{\mathcal{C}} \cup (\boldsymbol{I}^{\rightarrow}_{(\mathcal{C}+A)+B} \setminus \boldsymbol{I}^{\rightarrow}_{\mathcal{C}}) \cup (\boldsymbol{I}^{\leftarrow}_{(\mathcal{C}+A)+B} \setminus \boldsymbol{I}^{\leftarrow}_{\mathcal{C}}),\end{aligned}$$

where the last step is warranted by the observation that $\boldsymbol{S}^{\rightarrow} \subseteq \boldsymbol{T}^{\rightarrow}$ and $\boldsymbol{S}^{\leftarrow} \subseteq \boldsymbol{T}^{\leftarrow}$, for any two sets $\boldsymbol{S}$ and $\boldsymbol{T}$, where $\boldsymbol{S} \subseteq \boldsymbol{T}$.

Now, as $\boldsymbol{I}_{(\mathcal{C}+A)+B} = \boldsymbol{I}_{(\mathcal{C}+B)+A}$, we have:

$$\begin{aligned}&\boldsymbol{O}_{\mathcal{C}} \cup (\boldsymbol{I}^{\rightarrow}_{(\mathcal{C}+A)+B} \setminus \boldsymbol{I}^{\rightarrow}_{\mathcal{C}}) \cup (\boldsymbol{I}^{\leftarrow}_{(\mathcal{C}+A)+B} \setminus \boldsymbol{I}^{\leftarrow}_{\mathcal{C}})\\
=&\boldsymbol{O}_{\mathcal{C}} \cup (\boldsymbol{I}^{\rightarrow}_{(\mathcal{C}+B)+A} \setminus \boldsymbol{I}^{\rightarrow}_{\mathcal{C}}) \cup (\boldsymbol{I}^{\leftarrow}_{(\mathcal{C}+B)+A} \setminus \boldsymbol{I}^{\leftarrow}_{\mathcal{C}})\\
=&\boldsymbol{O}_{\mathcal{C}} \cup (\boldsymbol{I}^{\rightarrow}_{\mathcal{C}+B} \setminus \boldsymbol{I}^{\rightarrow}_{\mathcal{C}}) \cup (\boldsymbol{I}^{\leftarrow}_{\mathcal{C}+B} \setminus \boldsymbol{I}^{\leftarrow}_{\mathcal{C}})\\
&\cup (\boldsymbol{I}^{\rightarrow}_{(\mathcal{C}+B)+A} \setminus \boldsymbol{I}^{\rightarrow}_{\mathcal{C}+B}) \cup (\boldsymbol{I}^{\leftarrow}_{(\mathcal{C}+B)+A} \setminus \boldsymbol{I}^{\leftarrow}_{\mathcal{C}+B}) = \boldsymbol{O}_{(\mathcal{C}+B)+A}\end{aligned}$$

□

Thus, no matter in what order several arguments are moved from $\boldsymbol{U}_{\mathcal{C}}$ to $\boldsymbol{I}_{\mathcal{C}}$ and $\boldsymbol{O}_{\mathcal{C}}$, the resulting candidate is the same.

Now, we wish to use a tree of candidates as enumeration of preferred extensions. Given a candidate $\mathcal{C}$, we define a $\mathcal{C}$-tree inductively as follows:

- If $\boldsymbol{U}_{\mathcal{C}} = \varnothing$ then the tree consisting of the leaf $\mathcal{C}$ is a $\mathcal{C}$-tree.
- If $A \in \boldsymbol{U}_{\mathcal{C}} \setminus \text{refl}(\mathcal{C})$ then a tree with root node $\mathcal{C}$ having the roots of a $\mathcal{C}+A$-tree and a $\mathcal{C}-A$-tree as only children is a $\mathcal{C}$-tree.
- If $A \in \boldsymbol{U}_{\mathcal{C}} \cap \text{refl}(\mathcal{C})$ then a tree with root node $\mathcal{C}$ having the root of a $\mathcal{C}-A$-tree as only child is a $\mathcal{C}$-tree.

Example 4 ($\mathcal{C}$-trees): *We continue expanding on $\mathcal{C}_1$ as in Example 3. Repeated construction of candidates gives the $\mathcal{C}_1$-tree presented in Figure 1. Notice that some branches are shorter than others. This is because some additions to $\boldsymbol{I}$ imply additions to $\boldsymbol{O}$, and hence exhaust $\boldsymbol{U}$ sooner.*

Any tree, for which there is some candidate $\mathcal{C}$ such that the tree is a $\mathcal{C}$-tree, is called a *candidate tree*. The following results guarantee that candidate trees include all divisions that encode preferred extensions.

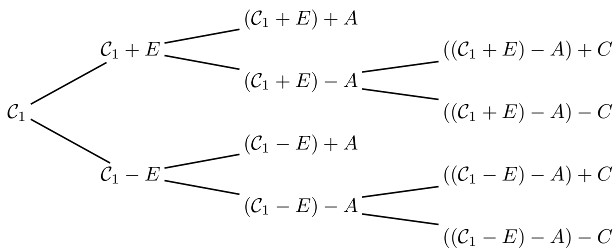

Figure 1. *A $\mathcal{C}_1$-tree*

Lemma 1. *Let $\boldsymbol{S} \subseteq \boldsymbol{A}$ be a conflict-free set, $\mathcal{C}$ a candidate, where $\boldsymbol{I}_\mathcal{C} \subseteq \boldsymbol{S}$ and $\boldsymbol{O}_\mathcal{C} \cap \boldsymbol{S} = \varnothing$, and A a member of $\boldsymbol{S} \setminus \boldsymbol{I}_\mathcal{C}$. Then $\mathcal{C} + A$ is a candidate, and $\boldsymbol{O}_{\mathcal{C}+A} \cap \boldsymbol{S} = \varnothing$.*

Proof. First note that A cannot be in refl$(\mathcal{C})$, as that would mean that there is a set $\boldsymbol{T} \subseteq \boldsymbol{I}_\mathcal{C}$ such that $\boldsymbol{T} \cup \{A\} \triangleright A$, which again would mean that $\boldsymbol{S}$ is not conflict-free. Therefore, Theorem 1 guarantees that $\mathcal{C} + A$ is a candidate, and we thus only need to show that $\boldsymbol{O}_{\mathcal{C}+A} \cap \boldsymbol{S} = \varnothing$.

As $\boldsymbol{O}_\mathcal{C} \cap \boldsymbol{S} = \varnothing$ it follows that $\boldsymbol{O}_{\mathcal{C}+A} \cap \boldsymbol{S} = (\boldsymbol{\Delta}^{\rightarrow}_{\mathcal{C}+A} \cup \boldsymbol{\Delta}^{\leftarrow}_{\mathcal{C}+A}) \cap \boldsymbol{S}$. If this set is non-empty, then there must be a B in $\boldsymbol{S}$, such that there is a set $\boldsymbol{T} \subseteq \boldsymbol{I}_\mathcal{C} \subseteq \boldsymbol{S}$ and element $C \in \boldsymbol{I}_\mathcal{C} \subseteq \boldsymbol{S}$, where either $\boldsymbol{T} \cup \{A\} \triangleright B$, $\boldsymbol{T} \cup \{B\} \triangleright A$, $\boldsymbol{T} \cup \{A, B\} \triangleright C$, or $\boldsymbol{T} \cup \{A, B\} \triangleright A$. But each of these imply that $\boldsymbol{S}$ is not conflict-free, and hence we conclude that $\boldsymbol{O}_{\mathcal{C}+A} \cap \boldsymbol{S} = \varnothing$. $\square$

Theorem 3. *Let $\mathcal{C}$ be a candidate, and $A \in \boldsymbol{U}_\mathcal{C}$. Then pref$(\mathcal{C}) =$ pref$(\mathcal{C} + A) \cup$ pref$(\mathcal{C} - A)$.*

Proof. It is obvious that pref$(\mathcal{C} + A) \cup$ pref$(\mathcal{C} - A) \subseteq$ pref$(\mathcal{C})$, so we only show that pref$(\mathcal{C}) \subseteq$ pref$(\mathcal{C} + A) \cup$ pref$(\mathcal{C} - A)$.

Let $\boldsymbol{S}^* \in$ pref$(\mathcal{C})$, i.e. $\boldsymbol{I}_\mathcal{C} \subseteq \boldsymbol{S}^* \subseteq \boldsymbol{I}_\mathcal{C} \cup \boldsymbol{U}_\mathcal{C}$. If A is not in $\boldsymbol{S}^*$, then it follows that $\boldsymbol{S}^* \subseteq \boldsymbol{I}_\mathcal{C} \cup \boldsymbol{U}_\mathcal{C} \setminus \{A\} = \boldsymbol{I}_{\mathcal{C}-A} \cup \boldsymbol{U}_{\mathcal{C}-A}$, and hence that $\boldsymbol{S}^* \in$ pref$(\mathcal{C} - A)$. If A is in $\boldsymbol{S}^*$ we similarly get that $\boldsymbol{S}^* \supseteq \boldsymbol{I}_\mathcal{C} \cup \{A\} = \boldsymbol{I}_{\mathcal{C}+A}$ and we only need to show that $\boldsymbol{S}^* \subseteq \boldsymbol{I}_{\mathcal{C}+A} \cup \boldsymbol{U}_{\mathcal{C}+A}$, i.e. that $\boldsymbol{O}_{\mathcal{C}+A} \cap \boldsymbol{S}^* = \varnothing$. But this is guaranteed by Lemma 1, and the result follows. $\square$

From this we immediately get:

Corollary 1. *If $\boldsymbol{S}^*$ is a preferred extension, then there is a leaf $\mathcal{C}$ of any $(\varnothing, \varnothing, \boldsymbol{A})$-tree, such that $\boldsymbol{S}^* \in$ pref$(\mathcal{C})$.*

Thus, when enumerating preferred extensions, it suffices to construct a single candidate tree, viz. a $(\varnothing, \varnothing, \boldsymbol{A})$-tree, even if candidates do not represent all possible divisions of $\boldsymbol{A}$. Furthermore, as the grounded extension of any system is a subset of any preferred extension [16,21], we have the following stronger result:

Corollary 2. *If $\boldsymbol{S}^*$ is a preferred extension, and $\boldsymbol{G}$ is the grounded extension, then there is a leaf $\mathcal{C}$ of any $(\boldsymbol{G}, \boldsymbol{G}^{\rightarrow} \cup \boldsymbol{G}^{\leftarrow}, \boldsymbol{A} \setminus (\boldsymbol{G} \cup \boldsymbol{G}^{\rightarrow} \cup \boldsymbol{G}^{\leftarrow}))$-tree, such that $\boldsymbol{S}^* \in$ pref$(\mathcal{C})$.*

4. Pruning of Candidate Trees

Depending on how a candidate tree is constructed, we might be able to prune it. In what follows we present some simple corollaries which allow for pruning of candidate trees.

Corollary 3. *Let $\mathcal{C}$ be a candidate for which pref$(\mathcal{C}) = \varnothing$. Then pref$(\mathcal{C}') = \varnothing$ for all nodes $\mathcal{C}'$ in any $\mathcal{C}$-tree.*

Thus, if during construction of a candidate tree, we create a candidate for which we know that pref$(\mathcal{C})$ is empty (e.g. by use of Theorems 6 or 7 below), then we do not have to construct the sub-tree rooted at that candidate.

Corollary 4. *Let $\mathcal{C}$ be a candidate. If $\boldsymbol{U}_\mathcal{C} = $ refl$(\mathcal{C})$, then pref$(\mathcal{C}) = $ pref$((\boldsymbol{I}_\mathcal{C}, \boldsymbol{O}_\mathcal{C} \cup \boldsymbol{U}_\mathcal{C}, \varnothing))$.*

Thus, if at some point in the construction of a candidate tree, we cannot find an argument to add to $\boldsymbol{I}_\mathcal{C}$, then we can stop exploring this branch of the tree.

Theorem 4. *Let $\mathcal{C}$ be a candidate. If $\boldsymbol{I}_\mathcal{C} \cup \boldsymbol{U}_\mathcal{C} \subsetneq \boldsymbol{S}^*$, for some admissible set $\boldsymbol{S}^*$, then pref$(\mathcal{C}) = \varnothing$.*

Proof. Obvious from Definition 2. □

Theorem 5. *Let $\mathcal{C}$ be a candidate. If $\boldsymbol{I}_\mathcal{C}^{\leftarrow} \setminus (\boldsymbol{I}_\mathcal{C} \cup \boldsymbol{U}_\mathcal{C})^{\rightarrow} \neq \varnothing$ then pref$(\mathcal{C}) = \varnothing$.*

Proof. Assume otherwise, and let $\boldsymbol{S}^* \in$ pref$(\mathcal{C})$ and $A \in \boldsymbol{I}_\mathcal{C}^{\leftarrow} \setminus (\boldsymbol{I}_\mathcal{C} \cup \boldsymbol{U}_\mathcal{C})^{\rightarrow}$. As $A \in \boldsymbol{I}_\mathcal{C}^{\leftarrow}$ it follows that there is some argument $B \in \boldsymbol{I}_\mathcal{C} \subseteq \boldsymbol{S}^*$ and set $\boldsymbol{T} \subseteq \boldsymbol{I}_\mathcal{C} \subseteq \boldsymbol{S}^*$, such that $\boldsymbol{T} \cup \{A\} \triangleright B$. Furthermore, as $\boldsymbol{S}^*$ is a preferred extension, it defends itself, and thus attacks some argument in $\boldsymbol{T} \cup \{A\}$. But as $\boldsymbol{S}^*$ is conflict-free, this argument must be A, and A must thus be in $\boldsymbol{S}^{*\rightarrow} \subseteq (\boldsymbol{I}_\mathcal{C} \cup \boldsymbol{U}_\mathcal{C})^{\rightarrow}$, which is a contradiction. □

Theorem 6. *Let $\mathcal{C}$ be a candidate and $A \in \boldsymbol{U}_\mathcal{C}$. If*

- *for all sets $\boldsymbol{T}$, where $\boldsymbol{T} \triangleright A$, it holds that $\boldsymbol{T} \cap \boldsymbol{I}_\mathcal{C}^{\rightarrow} \neq \varnothing$, and*
- *$A \notin (\boldsymbol{I}_\mathcal{C} \cup \boldsymbol{U}_\mathcal{C})^{\rightarrow}$, and*
- *$A \notin (\boldsymbol{I}_\mathcal{C} \cup \boldsymbol{U}_\mathcal{C} \setminus \{A\})^{\leftarrow}$*

then pref$(\mathcal{C} - A) = \varnothing$.

Proof. Assume that there is a $\boldsymbol{S}^* \in$ pref$(\mathcal{C} - A)$, i.e. that $\boldsymbol{I}_{\mathcal{C}-A} \subseteq \boldsymbol{S}^* \subseteq \boldsymbol{U}_{\mathcal{C}-A}$, which implies that $A \notin \boldsymbol{S}^*$. Hence, either $\boldsymbol{S}^*$ does not defend A, or $\boldsymbol{S}^* \cup \{A\}$ is not conflict-free. We show that both cases are impossible.

Let $\boldsymbol{T}$ be some minimal attack on A. Since we have that $\boldsymbol{T} \cap \boldsymbol{I}_\mathcal{C}^{\rightarrow} \neq \varnothing$, $\boldsymbol{I}_\mathcal{C}$ attacks $\boldsymbol{T}$, and hence that $\boldsymbol{S}^* \supseteq \boldsymbol{I}$ defends A, ruling out the first case.

If $\boldsymbol{S}^* \cup \{A\}$ is not conflict-free, but $\boldsymbol{S}^*$ is, then there is a set $\boldsymbol{T} \subseteq \boldsymbol{S}^* \subseteq (\boldsymbol{I}_\mathcal{C} \cup \boldsymbol{U}_\mathcal{C} \setminus \{A\})$ and argument $B \in \boldsymbol{S}^* \subseteq (\boldsymbol{I}_\mathcal{C} \cup \boldsymbol{U}_\mathcal{C} \setminus \{A\})$, such that either $\boldsymbol{T} \triangleright A$, $\boldsymbol{T} \cup \{A\} \triangleright A$, or $\boldsymbol{T} \cup \{A\} \triangleright B$. But the latter of these is precluded by $A \notin (\boldsymbol{I}_\mathcal{C} \cup \boldsymbol{U}_\mathcal{C} \setminus \{A\})^{\leftarrow}$ and the others by $A \notin (\boldsymbol{I}_\mathcal{C} \cup \boldsymbol{U}_\mathcal{C})^{\rightarrow}$. □

Theorem 7. *Let $\mathcal{C}$ be a candidate and $A \in \boldsymbol{U}_\mathcal{C}$ an argument, which is attacked by at least one set of arguments. If, for all pairs of sets $\boldsymbol{T}$ and $\boldsymbol{R}$, where $\boldsymbol{T} \triangleright \boldsymbol{R}$ and $\boldsymbol{R} \triangleright A$, it holds that $\boldsymbol{T} \cap \boldsymbol{O}_\mathcal{C} \neq \varnothing$, then pref$(\mathcal{C} + A) = \varnothing$.*

Proof. Assume $\boldsymbol{S}^* \in \text{pref}(\mathcal{C} + A)$, implying that $\boldsymbol{I}_{\mathcal{C}+A} \subseteq \boldsymbol{S}^*$, i.e. $A \in \boldsymbol{S}^*$. As $\boldsymbol{S}^*$ is a preferred extension, it must defend A. Let $\boldsymbol{R}$ be an attack on A (whose existence is guaranteed by the assumptions of the theorem). Since $\boldsymbol{S}^*$ defends A, it follows that there is a set $\boldsymbol{T} \subseteq \boldsymbol{S}^*$ such that $\boldsymbol{T} \triangleright \boldsymbol{R}$. But then $\boldsymbol{T}$ and $\boldsymbol{R}$ fulfills the conditions in the theorem, and $\boldsymbol{T} \cap \boldsymbol{O}_{\mathcal{C}} \neq \varnothing$. It follows that $\boldsymbol{S}^* \cap \boldsymbol{O}_{\mathcal{C}} \neq \varnothing$, which implies that $(\boldsymbol{I}_{\mathcal{C}} \cup \boldsymbol{U}_{\mathcal{C}}) \cap \boldsymbol{O}_{\mathcal{C}} \neq \varnothing$, which contradicts that $\mathcal{C}$ is a candidate. □

It may be possible to establish further pruning rules, especially for families of concrete argumentation systems, where the attack relation is known to abide by some restrictions. Moreover, it might be possible to establish heuristics for checking the conditions in the above theorems, or construct data structures which allow for these to be easily tested in $\mathcal{C} + A$ and $\mathcal{C} - A$ given the answers in $\mathcal{C}$. However, this is outside the scope of this paper.

As mentioned before, the method for answering questions about preferred extensions, presented here, is based on candidate trees. The exact nature of constructing/walking the trees we leave unspecified, as it may be dependent on the question that we seek an answer to and the system at hand. In some cases it may be suitable to use a depth-first walk of a candidate tree, and in others (such as when $|\boldsymbol{A}| = \infty$) a breath-first or iterated deepening depth-first walk will be needed. However, even though we leave out an exact specification of our method, we show how to apply it to an example:

Example 5 (Full-blown Example): *We round off the example system $\mathcal{A}_e$, presented in Example 2, by identifying all preferred extensions for it. As no sets of arguments are attacking A it is clear that it belongs to the grounded extension of $\mathcal{A}$. We therefore set out with constructing a $\mathcal{C}$-tree, where $\mathcal{C}$ is a candidate having $\boldsymbol{I}_{\mathcal{C}} = \{A\}$, such as $(\{A\}, \{F\}, \{B, C, D, E\})$. We construct the tree in a depth-first manner. The final result is shown in Figure 2.*

First we construct $\mathcal{C} + B = (\{A, B\}, \{C, D, F\}, \{E\})$ and then $(\mathcal{C} + B) + E = (\{A, B, E\}, \{C, D, F\}, \varnothing)$. Here $\{A, B, E\}$ is an admissible set, and $\boldsymbol{U}_{(\mathcal{C}+B)+E}$ is empty, so the recursion stops. Next we would need to consider $(\mathcal{C}+B)-E$, but $\mathcal{C}+B$ and E satisfies the conditions in Theorem 6 so we know that the sub-tree rooted at $(\mathcal{C}+B)-E$ contains no preferred extensions, so we skip it.

Instead we back-track and construct $\mathcal{C} - B = (\{A\}, \{B, F\}, \{C, D, E\})$, $(\mathcal{C} - B) + C = (\{A, C\}, \{B, F\}, \{D, E\})$, and then $((\mathcal{C}-B)+C)+D = (\{A, C, D\}, \{B, E, F\}, \varnothing)$. This latter one contains an admissible set, viz. $\{A, C, D\}$. Next, we construct $((\mathcal{C}-B)+C)-D = (\{A, C\}, \{B, D, F\}, \{E\})$, which satisfies the conditions in Theorem 5 (the satisfying element being B). Therefore, we do not investigate that sub-tree any further. Instead we back-track and construct $(\mathcal{C} - B) - C = (\{A\}, \{B, C, F\}, \{D, E\})$ and then $((\mathcal{C} - B) - C) + D = (\{A, D\}, \{B, C, E, F\}, \varnothing)$. Here $\{A, D\}$ is not a preferred extension (it does not attack B which attacks it). Back-tracking one level, we construct $((\mathcal{C} - B) - C) - D = (\{A\}, \{B, C, D, F\}, \{E\})$. This candidate satisfies the condition in Theorem 4, as $\{A, E\}$ is a subset of $\{A, B, E\}$, which we discovered previously.

The analysis thus shows that the two admissible sets of $\mathcal{A}$ having no admissible set as supersets (i.e. the preferred extensions), are $\{A, B, E\}$ and $\{A, C, D\}$.

Due to the restriction to candidates and the pruning rules, in the example we were able to deduce the result from five total divisions (out of 64 theoretically possible di-

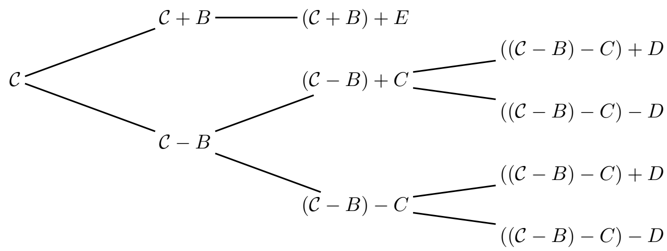

Figure 2. *Enumerating all preferred extensions of* $\mathcal{A}_e$.

visions), and with an overhead of five partial divisions. We think this is a satisfactory result, considering the highly intertwined nature of the example system. Of course, the *actual* efficiency of the method is influenced by a number of factors:

- How fast can the conditions in Theorems 4 to 7 be checked?
- In what order are candidates expanded. In the example above we went for exploring the largest sets as soon as possible, which allowed for ruling out sub-trees for smaller sets later on. Other heuristics may be better, depending on the problem being solved.

5. Conclusions

We have presented a method for enumerating the preferred extensions of argumentation system where joint attacks are allowed. We have proved that the method is complete and have presented a number of optimisation rules which should help reduce the running time of implementations. We do not claim that the set of these optimisation rules is complete, and acknowledge that details regarding implementation are still open for optimisation.

References

[1] C. Cayrol. On the relation between argumentation and non-monotonic coeherence-based entailment. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1443–1448. Morgan Kaufmann, 1995.

[2] R. Loui. Defeat among arguments: a system of defeasible inference. *Computational Intelligence*, 3:100–106, 1987.

[3] M. Elvang-Gøransson and A. Hunter. Argumentative logics: reasoning with classically inconsistent information. *Data and Knowledge Engineering*, 16:125–145, 1995.

[4] M. Elvang-Gøransson, P. Krause, and J. Fox. Dialectic reasoning with inconsistent information. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 114–121. Morgan Kaufmann, 1993.

[5] A. Bondarenko, P. M. Dung, R. A. Kowalski, and F. Toni. An abstract argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93:63–101, 1997.

[6] F. Lin. An argument-based approach to non-monotonic reasoning. *Computational Intelligence*, 9:254–267, 1993.

[7] H. Prakken and G. Vreeswijk. Logics for defeasible argumentation. In *Handbook of Philosophical Logic*. Kluwer Academic Publishers, 2000.

[8] S. Benferhat, D. Dubois, and H. Prade. Argumentative inference in uncertain and inconsistent knowledge bases. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 411–419. Morgan Kaufmann, 1993.

[9] J. Kohlas. Symbolic evidence, arguments, supports and valuation networks. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 186–198. Springer Verlag, 1993.

[10] P. Krause, S. Ambler, M. Elvang-Gøransson, and J. Fox. A logic of argumentation for reasoning under uncertainty. *Computational Intelligence*, 11:113–131, 1995.

[11] D. V. Carbogim, D. Robertson, and J. Lee. Argument-based applications to knowledge engineering. *The Knowledge Engineering Review*, 15(2), 2000.

[12] S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261—292, 1998.

[13] H. Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese*, 127:187–219, 2001.

[14] L. Amgoud, N. Maudet, and S. Parsons. An argumentation-based semantics for agent communication languages. In *Proceedings of the Fifteenth European Conference on Artificial Intelligence*, 2002.

[15] P. McBurney. *Rational Interaction*. PhD thesis, Department of Computer Science, University of Liverpool, 2002.

[16] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

[17] L. Amgoud. *Contribution a l'integration des préferences dans le raisonnement argumentatif*. PhD thesis, Université Paul Sabatier, July 1999.

[18] L. Amgoud and C. Cayrol. On the acceptability of arguments in preference-based argumentation framework. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 1–7, 1998.

[19] L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In *Proceedings of the Fourth International Conference on Multi-Agent Systems*, pages 31–38. IEEE Press, 2000.

[20] S. Parsons, M. Wooldridge, and L. Amgoud. Properties and complexity of formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3):347–376, 2003.

[21] S. H. Nielsen and S. Parsons. A generalization of Dung's abstract framework for argumentation. In *Proceedings of Third Workshop on Argumentation in Multi-agent Systems*. Springer, 2006.

[22] Alexander Bochman. Collective argumentation and disjunctive logic programming. *Journal of Logic and Computation*, 13(3):406–428, 2003.

[23] Y. Dimopoulos, B. Nebel, and F. Toni. Preferred arguments are harder to compute than stable extension. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 36–43. Morgan Kaufmann, 1999.

[24] S. Doutre and J. Mengin. Preferred extensions of argumentation frameworks: Query, answering and computation. In *Proceedings of International Joint Conference on Automated Reasoning*, volume 2083, pages 272–288. Springer, 2001.

[25] G. Vreeswijk and H. Prakken. Credulous and sceptical argument games for preferred semantics. In *Proceedings of the Seventh European Workshop on Logic in Artificial Inteligence, JELIA*, volume 1919, pages 239–253. Springer, 2000.

[26] C. Cayrol, S. Doutre, and J. Mengin. Dialectical proof theories for the credulous preferred semantics of argumentation frameworks. In *Proceedings of Sixth European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 668–679. Springer, 2001.