# Refining SCC decomposition in argumentation semantics: a first investigation

**Pietro Baroni** and **Massimiliano Giacomin**
Dipartimento di Elettronica per l'Automazione, University of Brescia
Via Branze 38
25123 Brescia Italy
{baroni,giacomin}@ing.unibs.it

**Abstract**

In the recently proposed SCC-recursive approach to argumentation semantics, the strongly connected components of an argumentation framework are used as the basic elements for the incremental construction of extensions. In this paper we argue that a finer decomposition, considering some suitably defined internal substructures of strongly connected components, called *autonomous fragments*, may be appropriate and support, in some cases, more intuitive results than the original approach. We cast this proposal within the SCC-recursive framework, show that it satisfies some fundamental requirements and provide some examples of its potential advantages.

## Introduction

The notion of SCC-recursiveness has recently been introduced (Baroni, Giacomin, & Guida 2005; Baroni & Giacomin 2004a) as a general scheme for argumentation semantics. On one hand, SCC-recursiveness is able to encompass most significant existing proposals such as *grounded semantics* (Pollock 1992) and *preferred semantics* (Dung 1995), on the other hand, it provides a sound basis for the definition and investigation of novel semantics proposals. In particular, the SCC-recursive $CF2$ semantics (Baroni & Giacomin 2004b; Baroni, Giacomin, & Guida 2005) has been shown to produce intuitively plausible results in some cases, involving odd-length cycles, which are quite problematic for other semantics. The definition of SCC-recursiveness stands on some widely accepted basic principles which can be regarded as a common ground for any argumentation semantics: the *conflict free* principle, the *reinstatement* principle (Prakken & Vreeswijk 2001), and the *directionality* principle. In particular, the last one suggests that defeat dependencies among arguments can be taken into account following the partial order induced by the decomposition of the graph representation of an argumentation framework into *Strongly Connected Components* (SCCs). In some cases it emerges however that this decomposition is, in a sense, still "too rough" to capture some intuitively significant aspects of the defeat graph topology and a further decomposition may be appropriate. This work starts from this observation and presents a preliminary investigation about why and how such a finer decomposition can be carried out in the framework of SCC-recursive semantics. The paper is organized as follows. In the following section we recall the necessary background concepts on SCC-recursiveness, while in the next one we introduce some motivating examples for our investigation. We then introduce the notion of *autonomous fragments* within a strongly connected component and show how this notion can be exploited within the SCC-recursive scheme. After exemplifying the application of the proposed approach, we conclude the paper with some final remarks.

## SCC-recursiveness

We first give an account of SCC-recursiveness as introduced in (Baroni, Giacomin, & Guida 2005). The approach lies in the frame of the general theory of abstract argumentation frameworks proposed by Dung (Dung 1995).

**Definition 1** *An argumentation framework is a pair* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$*, where* $\mathcal{A}$ *is a set, and* $\rightarrow \subseteq (\mathcal{A} \times \mathcal{A})$ *is a binary relation on* $\mathcal{A}$*, called attack relation.*

In the following we will always assume that $\mathcal{A}$ is finite. An argumentation framework $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$ can be represented as a directed graph, called *defeat graph*, where nodes are the arguments and edges correspond to the elements of the attack relation. In the following, the nodes that attack a given argument $\alpha$ are called *defeaters* of $\alpha$ and form a set which is denoted as $\mathrm{parents}_{\mathrm{AF}}(\alpha)$:

**Definition 2** *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$ *and a node* $\alpha \in \mathcal{A}$*, we define* $\mathrm{parents}_{\mathrm{AF}}(\alpha) = \{\beta \in \mathcal{A} \mid \beta \rightarrow \alpha\}$*. If* $\mathrm{parents}_{\mathrm{AF}}(\alpha) = \emptyset$*, then* $\alpha$ *is called an* initial *node.*

Since we will frequently consider properties of sets of arguments, it is useful to extend to them the notations defined for the nodes:

**Definition 3** *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$*, a node* $\alpha \in \mathcal{A}$ *and two sets* $S, P \subseteq \mathcal{A}$*, we define:*

$$S \rightarrow \alpha \equiv \exists \beta \in S : \beta \rightarrow \alpha$$
$$\alpha \rightarrow S \equiv \exists \beta \in S : \alpha \rightarrow \beta$$
$$S \rightarrow P \equiv \exists \alpha \in S, \beta \in P : \alpha \rightarrow \beta$$
$$\mathrm{outparents}_{\mathrm{AF}}(S) = \{\alpha \in \mathcal{A} \mid \alpha \notin S \wedge \alpha \rightarrow S\}$$

In Dung's theory, an argumentation semantics is defined by specifying the criteria for deriving, given a generic argumentation framework, the set of all possible extensions, each one representing a set of arguments considered to be acceptable together. Typically an argument is considered *justified* if and only if it belongs to all extensions. Given a generic argumentation semantics $\mathcal{S}$, the set of extensions prescribed by $\mathcal{S}$ for a given argumentation framework $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$ is denoted as $\mathcal{E}_{\mathcal{S}}(\mathrm{AF})$. If it holds that $\forall \mathrm{AF}, |\mathcal{E}_{\mathcal{S}}(\mathrm{AF})| = 1$, then the semantics $\mathcal{S}$ is said to follow the *unique-status approach*, otherwise it is said to follow the *multiple-status approach* (Prakken & Vreeswijk 2001).

SCC-recursiveness is a property of the extensions which relies on the graph theoretical notion of *strongly connected components* (SCCs).

**Definition 4** *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$*, the binary relation of path-equivalence between nodes, denoted as* $PE_{\mathrm{AF}} \subseteq (\mathcal{A} \times \mathcal{A})$*, is defined as follows:*

- $\forall \alpha \in \mathcal{A}, (\alpha, \alpha) \in PE_{\mathrm{AF}}$
- *given two distinct nodes* $\alpha, \beta \in \mathcal{A}$*,* $(\alpha, \beta) \in PE_{\mathrm{AF}}$ *if and only if there is a path from* $\alpha$ *to* $\beta$ *and a path from* $\beta$ *to* $\alpha$*.*

The *strongly connected component* of AF are the equivalence classes of nodes under the relation of path-equivalence. The set of the SCCs of AF is denoted as $\mathrm{SCCS}_{\mathrm{AF}}$. A particular case is represented by the empty argumentation framework: when $\mathrm{AF} = \langle \emptyset, \emptyset \rangle$ we assume $\mathrm{SCCS}_{\mathrm{AF}} = \{\emptyset\}$.

We extend to SCCs the notion of parents, namely the set of the other SCCs that attack a SCC $S$, which is denoted as $\mathrm{sccpar}_{\mathrm{AF}}(S)$, and we introduce the definition of *proper ancestors*, denoted as $\mathrm{sccanc}_{\mathrm{AF}}(S)$:

**Definition 5** *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$ *and a SCC* $S \in \mathrm{SCCS}_{\mathrm{AF}}$*, we define*

$$\mathrm{sccpar}_{\mathrm{AF}}(S) = \{P \in \mathrm{SCCS}_{\mathrm{AF}} \mid P \neq S \text{ and } P \rightarrow S\}$$

*and*

$$\mathrm{sccanc}_{\mathrm{AF}}(S) = \mathrm{sccpar}_{\mathrm{AF}}(S) \cup \bigcup_{P \in \mathrm{sccpar}_{\mathrm{AF}}(S)} \mathrm{sccanc}_{\mathrm{AF}}(P)$$

A SCC $S$ such that $\mathrm{sccpar}_{\mathrm{AF}}(S) = \emptyset$ is called *initial*.

It is well-known that the graph obtained by considering SCCs as single nodes is acyclic. In other words, SCCs can be partially ordered according to the relation of attack. Following the above definition, *initial* SCCs are those which are not preceded by any other one in this partial order. Of course, in any argumentation framework there is at least one initial SCC. This fact lies at the heart of the definition of SCC-recursiveness, which is based on the intuition that extensions can be built incrementally starting from initial SCCs and following the above mentioned partial order. In other words, the choices concerning extension construction carried out in an initial SCC do not depend on those concerning the other ones, while they directly affect the choices about the subsequent SCCs and so on.

While the basic underlying intuition is rather simple, the formalization of SCC-recursiveness is admittedly rather complex and involves some additional notions. Due to space limitations, we can only give here a quick account, while referring the reader to (Baroni, Giacomin, & Guida 2005) for more details and examples. First of all, the choices in the antecedent SCCs determine a partition of the nodes of a generic SCC $S$ into three subsets:

**Definition 6** *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$*, a set* $E \subseteq \mathcal{A}$ *and a set* $S \subseteq \mathcal{A}$*, we define:*

- $D_{\mathrm{AF}}(S, E) = \{\alpha \in S \mid (E \cap \mathrm{outparents}_{\mathrm{AF}}(S)) \rightarrow \alpha\}$
- $P_{\mathrm{AF}}(S, E) = \{\alpha \in S \mid (E \cap \mathrm{outparents}_{\mathrm{AF}}(S)) \not\rightarrow \alpha \wedge \exists \beta \in (\mathrm{outparents}_{\mathrm{AF}}(S) \cap \mathrm{parents}_{\mathrm{AF}}(\alpha)) : E \not\rightarrow \beta \wedge \alpha \not\rightarrow \beta\}$
- $U_{\mathrm{AF}}(S, E) = S \setminus (D_{\mathrm{AF}}(S, E) \cup P_{\mathrm{AF}}(S, E)) = \{\alpha \in S \mid (E \cap \mathrm{outparents}_{\mathrm{AF}}(S)) \not\rightarrow \alpha \wedge \forall \beta \in (\mathrm{outparents}_{\mathrm{AF}}(S) \cap \mathrm{parents}_{\mathrm{AF}}(\alpha))\ E \cup \{\alpha\} \rightarrow \beta\}$

Definition 6 is a generalized version (useful for the sequel of the paper) of the corresponding Definition 18 of (Baroni, Giacomin, & Guida 2005). In words, the set $D_{\mathrm{AF}}(S, E)$ consists of the nodes of $S$ attacked by $E$ from outside $S$, the set $U_{\mathrm{AF}}(S, E)$ includes any node $\alpha$ of $S$ that is not attacked by $E$ from outside $S$ and is defended by $E$ or defends itself (i.e. the defeaters of $\alpha$ from outside $S$ are all attacked by $E$ or by $\alpha$ itself), and $P_{\mathrm{AF}}(S, E)$ includes any node $\alpha$ of $S$ that is not attacked by $E$ from outside $S$ and is not defended by $E$ or by itself (i.e. at least one of the defeaters of $\alpha$ from outside $S$ is not attacked by $E$ nor by $\alpha$). It is easy to verify that, when $S$ is a SCC, as in the original Definition 18 of (Baroni, Giacomin, & Guida 2005), $D_{\mathrm{AF}}(S, E)$, $P_{\mathrm{AF}}(S, E)$ and $U_{\mathrm{AF}}(S, E)$ are determined only by the elements of $E$ that belong to the SCCs in $\mathrm{sccanc}_{\mathrm{AF}}(S)$ and it may not be the case that a node $\alpha \in S$ defends itself against an attack coming from outside $S$. Regarding $E$ as a part of an extension which is being constructed, the idea is then that arguments in $D_{\mathrm{AF}}(S, E)$, being attacked by nodes in $E$, cannot be chosen in the construction of the extension $E$ (i.e. do not belong to $E \cap S$). Selection of arguments to be included in $E$ is therefore restricted to $(S \setminus D_{\mathrm{AF}}(S, E)) = (U_{\mathrm{AF}}(S, E) \cup P_{\mathrm{AF}}(S, E))$, which, for ease of notation, will be denoted in the following as $UP_{\mathrm{AF}}(S, E)$.

To formalize this aspect, we define the *restriction* of an argumentation framework to a given subset of its nodes:

**Definition 7** *Let* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$ *be an argumentation framework, and let* $S \subseteq \mathcal{A}$ *be a set of arguments. The* restriction *of* AF *to* $S$ *is the argumentation framework* $\mathrm{AF}{\downarrow}_S = \langle S, \rightarrow \cap (S \times S) \rangle$*.*

Inspired by the reinstatement principle, we require the selection of nodes within a SCC $S$ to be carried out on the restricted argumentation framework $\mathrm{AF}{\downarrow}_{UP_{\mathrm{AF}}(S,E)}$ without taking into account the attacks coming from $D_{\mathrm{AF}}(S, E)$.

Combining these ideas and skipping some details not strictly necessary in the context of the present paper, we can finally recall the definition of *SCC-recursiveness*:

**Definition 8** *A given argumentation semantics* $\mathcal{S}$ *is* SCC-recursive *if and only if for any argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$*,* $\mathcal{E}_{\mathcal{S}}(\mathrm{AF}) = \mathcal{GF}(\mathrm{AF}, \mathcal{A})$*, where for any* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$ *and for any set* $C \subseteq \mathcal{A}$*, the function* $\mathcal{GF}(\mathrm{AF}, C) \subseteq 2^{\mathcal{A}}$ *is defined as follows:*
*for any* $E \subseteq \mathcal{A}$*,* $E \in \mathcal{GF}(\mathrm{AF}, C)$ *if and only if*

- *in case* $|\mathrm{SCCS}_{\mathrm{AF}}| = 1$*,* $E \in \mathcal{BF}_{\mathcal{S}}(\mathrm{AF}, C)$

- *otherwise*, $\forall S \in \mathrm{SCCS}_{\mathrm{AF}}$
  $(E \cap S) \in \mathcal{GF}(\mathrm{AF}\downarrow_{UP_{\mathrm{AF}}(S,E)}, U_{\mathrm{AF}}(S,E) \cap C)$

where $\mathcal{BF}_{\mathcal{S}}(\mathrm{AF}, C)$ is a function, called *base function*, that, given an argumentation framework $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$ such that $|\mathrm{SCCS}_{\mathrm{AF}}| = 1$ and a set $C \subseteq \mathcal{A}$, gives a subset of $2^{\mathcal{A}}$.

Since this definition is somewhat arduous to examine in its full detail, we just give some "quick and dirty" indications which are useful for the sequel of the paper (in particular, we do not consider the meaning of the parameter $C$ in the description, as not necessary for the comprehension of this paper). The set of extensions $\mathcal{E}_{\mathcal{S}}(\mathrm{AF})$ of an argumentation framework AF is given by $\mathcal{GF}(\mathrm{AF}, \mathcal{A})$, namely by the invocation of the function $\mathcal{GF}$ which receives as parameters an argumentation framework (in this case the whole AF) and a set of arguments (in this case the whole $\mathcal{A}$). The function $\mathcal{GF}(\mathrm{AF}, C)$ is defined recursively. The base of the recursion is reached when AF consists of a unique SCC: in this case the set of extensions is directly given by the invocation of a semantic-specific base function $\mathcal{BF}_{\mathcal{S}}(\mathrm{AF}, C)$. In the other case, for each SCC $S$ of AF the function $\mathcal{GF}$ is invoked recursively on the restriction $\mathrm{AF}\downarrow_{UP_{\mathrm{AF}}(S,E)}$.

Note that the restriction concerns $UP_{\mathrm{AF}}(S, E)$, namely the part of $S$ which "survives" the attacks of the preceding ones in the partial order. The definition also has a constructive interpretation, which suggests an effective (recursive) procedure for computing all the extensions of an argumentation framework $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$ once a specific base function characterizing the semantics is assigned. A particular role in this context is played by the initial SCCs. In fact, for any initial SCC $I$, since by definition there are no outer attacks, the set of defended nodes coincides with $I$, i.e. $UP_{\mathrm{AF}}(I, E) = U_{\mathrm{AF}}(I, E) = I$ for any $E$. This gives rise to the invocation $\mathcal{GF}(\mathrm{AF}\downarrow_I, I)$ for any initial SCC $I$. Since $\mathrm{AF}\downarrow_I$ obviously consists of a unique SCC, according to Definition 8 the base function $\mathcal{BF}_{\mathcal{S}}(\mathrm{AF}\downarrow_I, I)$ is invoked, which returns the extensions of $\mathrm{AF}\downarrow_I$ according to the semantics $\mathcal{S}$. Therefore, the base function can be first computed on the initial SCCs, where it directly returns the extensions prescribed by the semantics. Then, the results of this computation are used to identify, within the subsequent SCCs, the restricted argumentation frameworks on which the procedure is recursively invoked.

All SCC-recursive semantics "share" this general scheme and only differ by the specific base function adopted. It has been shown that all semantics encompassed by Dung's framework are SCC-recursive and the relevant base functions have been identified. Among them, in the following we will mainly refer to grounded semantics (denoted as $\mathcal{GR}$) and preferred semantics (denoted as $\mathcal{PR}$) considered the "best" representatives of the unique-status and multiple-status approach respectively.

Moreover, defining and experimenting new SCC-recursive semantics is quite easy since it simply amounts to defining a base function operating on single-SCC argumentation frameworks. As shown in (Baroni, Giacomin, & Guida 2005), the base function has only to respect two very simple conditions in order to ensure that the resulting extensions satisfy the fundamental requirements of being conflict-free and of agreement with grounded semantics.

As to the conflict-free property, it is sufficient that the base function returns only conflict-free subsets.

**Definition 9** *A semantics $\mathcal{S}$ satisfies the* conflict-free property *if and only if* $\forall \mathrm{AF}, \forall E \in \mathcal{E}_{\mathcal{S}}(\mathrm{AF})$ *$E$ is conflict-free.*

**Definition 10** *The base function of a SCC-recursive semantics $\mathcal{S}$ is* conflict-free *if and only if* $\forall \mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$ *and* $\forall C \subseteq \mathcal{A}$ *each element of* $\mathcal{BF}_{\mathcal{S}}(\mathrm{AF}, C)$ *is conflict-free.*

**Proposition 1** *(Theorem 48 of (Baroni, Giacomin, & Guida 2005)) Given a SCC-recursive semantics $\mathcal{S}$, if its base function is conflict-free then $\mathcal{S}$ satisfies the conflict-free property.*

As to the agreement with grounded semantics, it is sufficient that the base function properly deals just with the simplest case of non-empty argumentation framework (a single node not attacking itself).

**Proposition 2** *(Theorem 52 of (Baroni, Giacomin, & Guida 2005)) Let $\mathcal{S}$ be a SCC-recursive semantics identified by a conflict-free base function such that*

$$\mathcal{BF}_{\mathcal{S}}(\langle \{\alpha\}, \emptyset \rangle, \{\alpha\}) = \{\{\alpha\}\}$$

*For any argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$, $\forall E \in \mathcal{E}_{\mathcal{S}}(\mathrm{AF})$, *the grounded extension* $\mathrm{GE}(\mathrm{AF}) \subseteq E$.

Thanks to these properties, four original SCC-recursive semantics have been defined in (Baroni, Giacomin, & Guida 2005) in a relatively straightforward way. In particular, the SCC-recursive semantics called $CF2$ (Baroni & Giacomin 2003; Baroni, Giacomin, & Guida 2005) has been shown to provide a good behavior in several critical examples, while featuring a very simple base function: $\mathcal{BF}_{CF2}(\mathrm{AF}, C) = \mathcal{MCF}_{\mathrm{AF}}$, where $\mathcal{MCF}_{\mathrm{AF}}$ denotes the set made up of all the maximal conflict-free sets of AF (note that the parameter $C$ plays no role at all in this case).

## Motivating examples

In the SCC-recursive approach, SCCs play the role of basic decomposition elements on which the semantics-specific base function is applied. In $CF2$ semantics, as well as in the SCC-recursive formulation of grounded, stable, and preferred semantics, the base function does not take into account the "internal topology" of the SCC to which it is applied. Roughly speaking, since all elements of a SCC are mutually reachable, it has been implicitly assumed in (Baroni, Giacomin, & Guida 2005) that they can be treated as "equivalent" in the construction of extensions.

Though this hidden assumption is reasonable in most situations, there are cases where it can be regarded as questionable.

As a first example, consider the argumentation framework $\mathrm{AF}_1$ represented in Figure 1. $\mathrm{AF}_1$ clearly consists of a single SCC, so its extensions in a SCC-recursive semantics are directly obtained by applying the base function to the whole $\mathrm{AF}_1$. In the case of $CF2$ semantics it turns out that $\mathcal{E}_{CF2}(\mathrm{AF}_1) = \mathcal{MCF}_{\mathrm{AF}_1} = \{\{\beta\}, \{\gamma\}\}$. According to both grounded and preferred semantics, the only extension in this case is the empty set: $\mathcal{E}_{\mathcal{PR}}(\mathrm{AF}_1) = \mathcal{E}_{\mathcal{GR}}(\mathrm{AF}_1) =$

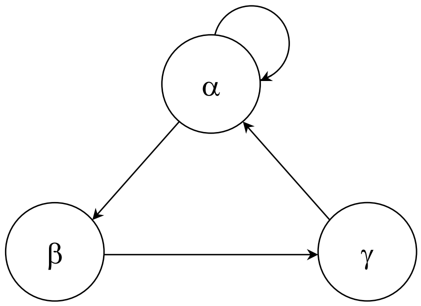


Figure 1: A self-defeating node within a three-length cycle ($\mathrm{AF}_1$).

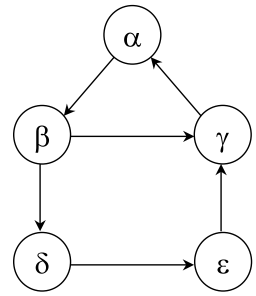


Figure 2: A three-length cycle within a five-length cycle($\mathrm{AF}_2$).

$\{\emptyset\}$. Therefore in all these semantics no argument is included in all the prescribed extensions and can be considered justified. However, this result can be questioned. In fact, one may object that the node $\alpha$, being self-defeating, is intrinsically weak and should not be able to affect the justification status of the arguments it attacks. In this perspective, $\alpha$ should be ruled out, $\beta$, not receiving attacks anymore, should be regarded as justified and, as a consequence, $\gamma$ should not be justified. While this kind of behavior is not supported by any of the above mentioned semantics, it could be obtained by a sort of ad-hoc rule or by some form of graph preprocessing devoted to suppress all self-defeating nodes.

Other examples call however for a more general approach to this kind of situation. Consider the argumentation framework $\mathrm{AF}_2$ represented in Figure 2, which also consists of a unique SCC. According to $CF2$ semantics we have $\mathcal{E}_{CF2}(\mathrm{AF}_2) = \mathcal{MCF}_{\mathrm{AF}_2} = \{\{\alpha,\delta\},\{\alpha,\epsilon\},\{\gamma,\delta\},\{\beta,\epsilon\}\}$, while $\mathcal{E}_{\mathcal{PR}}(\mathrm{AF}_2) = \mathcal{E}_{\mathcal{GR}}(\mathrm{AF}_2) = \{\emptyset\}$. Again, no argument is justified according to any of the above semantics. While this may sound very reasonable, other interpretations are possible, depending on the meaning ascribed to odd-length cycles. In fact, it can be noted that arguments $\alpha$, $\beta$, and $\gamma$ form a three length cycle, independently of $\delta$ and $\epsilon$. $CF2$ semantics is based on the idea that even- and odd-length cycles share the same nature and should be treated equally (Baroni & Giacomin 2003) in a "length-independent" way. However, as pointed out in (Prakken & Vreeswijk 2001), a different point view is also possible where "odd defeat loops are of an essentially different kind than even defeat loops". For instance, one might state that odd-length cycles are like paradoxes, i.e. situations where nothing can be believed, while even-length cycles are like dilemmas, i.e. situations where a choice needs to be made. According to this view, the arguments $\alpha$, $\beta$, and $\gamma$ should in a sense "annul" each other and lose the power of affecting other nodes, leaving then $\delta$ undefeated and, consequently, $\epsilon$ defeated. It has to be acknowledged that the choice of the "most appropriate" result is a matter of debate and may depend on case-specific considerations too (Horty 2002) and that some critical examples may be dealt with by applying some rationality postulates at the level of argument structure and construction (Caminada & Amgoud 2005). At an abstract level, it is however desirable that the alternative view presented above can be encompassed within the general SCC-recursive scheme. As it emerges from the above examples, this requires, first of all, the capability to distinguish some significant substructures (the self-defeating node in the first case, the three length-cycle in the second case) within a single SCC: this aspect is dealt with in the next section.

## Decomposing a SCC into autonomous fragments

We follow the idea of identifying, within a SCC $S$, the subsets of nodes that can be considered "autonomously" in the incremental construction of extensions. These subsets will be called *autonomous fragments* and the set of autonomous fragments of $S$ will be denoted as $\mathcal{AU}(S)$. A first intuitive requirement is that each fragment is strongly connected by itself and, while respecting this property, is as small as possible. Moreover, to be autonomous, fragments should not "interfere", namely should not intersect each other. To define significant minimal strongly connected fragments within a "conventional" SCC we need to modify the definition of path equivalence, substituting the clause that each node is always path-equivalent to itself with the requirement that the node is self-defeating.

**Definition 11** *Given an argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$ *and a set* $Q \subseteq \mathcal{A}$*, let* $\mathrm{AF}' = \mathrm{AF}{\downarrow}_Q$*. The binary relation of path-mutuality restricted to* $Q$*, denoted as* $PM_Q \subseteq (Q \times Q)$*, is defined as follows:*

- $\forall \alpha \in Q, (\alpha,\alpha) \in PM_Q$ *if and only if* $(\alpha,\alpha) \in \rightarrow$*;*
- *given two distinct nodes* $\alpha,\beta \in Q$*,* $(\alpha,\beta) \in PM_Q$ *if and only if in* $\mathrm{AF}'$ *there is a path from* $\alpha$ *to* $\beta$ *and a path from* $\beta$ *to* $\alpha$*.*

We define the notion of *fragments* of a SCC $S$ as follows.

**Definition 12** *Given a non-empty argumentation framework* $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$ *and a SCC* $S \in \mathrm{SCCS}_{\mathrm{AF}}$*, a non-empty set* $F \subseteq S$ *is called a fragment of* $S$ *if and only if* $\forall \alpha,\beta \in F$*,*

$(\alpha, \beta) \in PM_F$. *The set of fragments of* $S$ *is denoted as* $\mathcal{FR}(S)$.

Note that $\alpha$ and $\beta$ are not necessarily distinct in Definition 12 and that the fragments belonging to $\mathcal{FR}(S)$ generally intersect each other. In particular, $S \in \mathcal{FR}(S)$ unless $S$ consists of a unique non self-defeating argument, namely $S = \{\alpha\}$ and $\mathrm{AF}{\downarrow}_S = \langle\{\alpha\}, \emptyset\rangle$ (such a SCC will be called *monadic*). We have therefore the guarantee that $\mathcal{FR}(S) \neq \emptyset$ if $S$ is not monadic.

The set $\mathcal{AU}(S)$ of autonomous fragments of a non monadic SCC $S$ is derived from $\mathcal{FR}(S)$ by applying the following algorithm.

**Definition of algorithm** $\mathcal{AU}$

**Step 1**
`let` $\Sigma = \mathcal{FR}(S)$;
**BEGIN MAIN LOOP**
**Step 2**
`let` $\Sigma_{min} = \{F \in \Sigma \mid \nexists G \in \Sigma : G \subsetneq F\}$;
**Step 3**
`let` $\mathcal{AU}(S) = \{F \in \Sigma_{min} \mid \forall G \in \Sigma_{min} : G \neq F,$
$F \cap G = \emptyset\}$;
**Step 4**
`if` $\mathcal{AU}(S) \neq \emptyset$
`then`
  **EXIT**;
`else`
  `let` $\Sigma = \Sigma \setminus \Sigma_{min}$;
  `goto Step 2;`
`endif`
**END MAIN LOOP**

In Step 1 the variable $\Sigma$ is initialized to contain the set of all fragments of $S$. Then the algorithm enters a loop. In Step 2 the set $\Sigma_{min}$ of the elements of $\Sigma$ which are minimal with respect to set inclusion is identified, according to the intuition that autonomous fragments are as small as possible. In Step 3 the condition of non interference is verified, by selecting for inclusion into $\mathcal{AU}(S)$ the elements of $\Sigma_{min}$ which have empty intersection with any other element of $\Sigma_{min}$ (note that it may be the case that no such element exist in $\Sigma_{min}$ at a given iteration of the main loop). If a non-empty $\mathcal{AU}(S)$ has been identified in Step 3, then in Step 4 the algorithm terminates, otherwise all elements of $\Sigma_{min}$ are dropped from $\Sigma$ and a new iteration of the main loop begins.

**Proposition 3** *Let* $S$ *be a non monadic SCC of an argumentation framework* AF*, then algorithm* $\mathcal{AU}$ *is guaranteed to terminate by producing a non-empty set* $\mathcal{AU}(S)$.

Recall that, since $S$ is non monadic, $S \in \mathcal{FR}(S)$ and therefore $S \in \Sigma$ since the initialization of $\Sigma$ in Step 1. Moreover, the finiteness of $S$ ensures that $\Sigma$ is finite. Then two situations may occur. If it holds that $\Sigma = \{S\}$, then clearly $\Sigma_{min} = \{S\}$ is assigned in Step 2 and $\mathcal{AU}(S) = \{S\}$ is assigned in Step 3, which determines algorithm termination. Otherwise $\Sigma \supsetneq \{S\}$ and $S \notin \Sigma_{min}$ since the other elements of $\Sigma$ are proper subsets of $S$. Then two cases are possible: the algorithm terminates with a non-empty $\mathcal{AU}(S)$ or a new iteration begins after subtracting $\Sigma_{min}$ from $\Sigma$. In the new iteration it still holds that $S \in \Sigma$ and we can iterate the same reasoning: then, the finiteness of $\Sigma$ ensures that one of the two termination cases considered above is reached in a finite number of iterations. □

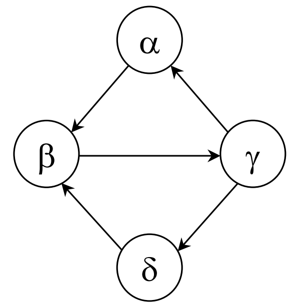


Figure 3: Two three-length cycles within a four-length cycle($\mathrm{AF}_3$).

To complete the definition of $\mathcal{AU}(S)$, the cases of a monadic SCC and of an empty SCC (which occurs only in an empty argumentation framework) have to be covered.

**Definition 13** *Given an argumentation framework* $\mathrm{AF} = \langle\mathcal{A}, \rightarrow\rangle$ *and a SCC* $S \in \mathrm{SCCS}_{\mathrm{AF}}$*, the set of autonomous fragments of* $S$*, denoted as* $\mathcal{AU}(S)$*, is defined as follows:*

- $\mathcal{AU}(S) = \{S\}$, if $S$ is monadic or $S = \emptyset$;
- $\mathcal{AU}(S)$ is the result of applying algorithm $\mathcal{AU}$ to $S$, otherwise.

By inspection of Step 3 of algorithm $\mathcal{AU}$, it can be noted that the elements of $\mathcal{AU}(S)$ are disjoint.

Let us now examine some examples of application of algorithm $\mathcal{AU}$. Every argumentation framework $\mathrm{AF}_i$ considered in the following consists of a single SCC $S_i$ (i.e. $\mathrm{SCCS}_{\mathrm{AF}_i} = \{S_i\}$), which coincides with the set of all arguments of $\mathrm{AF}_i$. In the case of $\mathrm{AF}_1$ (Figure 1), $\mathcal{FR}(S_1) = \{\{\alpha\}, S_1\}$ and algorithm $\mathcal{AU}$ terminates in one iteration with $\mathcal{AU}(S_1) = \Sigma_{min} = \{\{\alpha\}\}$.

In the case of $\mathrm{AF}_2$ (Figure 2), $\mathcal{FR}(S_2) = \{\{\alpha, \beta, \gamma\}, S_2\}$ and algorithm $\mathcal{AU}$ terminates in one iteration with $\mathcal{AU}(S_2) = \Sigma_{min} = \{\{\alpha, \beta, \gamma\}\}$.

Consider now $\mathrm{AF}_3$ (Figure 3). $\mathcal{FR}(S_3) = \{\{\alpha, \beta, \gamma\}, \{\beta, \gamma, \delta\}, S_3\}$. Then in the first iteration of the main loop $\Sigma_{min} = \{\{\alpha, \beta, \gamma\}, \{\beta, \gamma, \delta\}\}$, and, since the intersection of the two elements of $\Sigma_{min}$ is not empty, $\mathcal{AU}(S) = \emptyset$ in Step 3 and $\Sigma = \{S_3\}$ results in the `else` branch of Step 4. In the subsequent iteration, the algorithm terminates with $\mathcal{AU}(S_3) = \{S_3\} = \{\{\alpha, \beta, \gamma, \delta\}\}$.

In $\mathrm{AF}_4$ (Figure 4), $\mathcal{FR}(S_4) = \{\{\alpha, \beta\}, \{\gamma, \delta\}, S_4\}$. Then in the first iteration of the main loop $\Sigma_{min} = \{\{\alpha, \beta\}, \{\gamma, \delta\}\}$ and, since the intersection of the two elements of $\Sigma_{min}$ is empty, the algorithm terminates with $\mathcal{AU}(S_4) = \{\{\alpha, \beta\}, \{\gamma, \delta\}\}$.

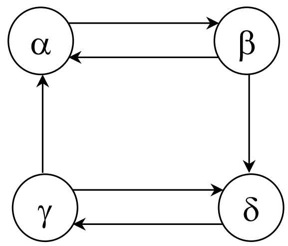


Figure 4: Two two-length cycles within a four-length cycle($\mathrm{AF}_4$).

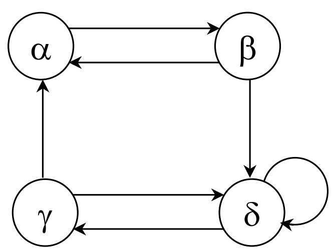


Figure 5: Two two-length cycles within a four-length cycle with a self-defeating node($\mathrm{AF}_5$).

In $\mathrm{AF}_5$ (Figure 5), $\mathcal{FR}(S_5) = \{\{\delta\}, \{\alpha, \beta\}, \{\gamma, \delta\}, S_5\}$. Then in the first iteration of the main loop $\Sigma_{min} = \{\{\alpha, \beta\}, \{\delta\}\}$ and, since the intersection of the two elements of $\Sigma_{min}$ is empty, the algorithm terminates with $\mathcal{AU}(S_5) = \{\{\alpha, \beta\}, \{\delta\}\}$.

## Exploiting autonomous fragments in SCC-recursive semantics

Let us now continue our investigation by looking for a way to take into account the notion of autonomous fragment within the SCC-recursive scheme. Since this notion is introduced at the level of single SCCs, the most direct way is considering it within the definition of the base function, which operates at this level. Each base function $\mathcal{BF}$ considered in (Baroni, Giacomin, & Guida 2005) directly selects a set of subsets of a SCC $S$. An *$\mathcal{AU}$-aware* base function (denoted in the following as $\mathcal{BF}'$) should instead take into account the autonomous fragments of $S$. Our intuition is that each autonomous fragment represents the minimal topological unit to which some semantics-specific principle can be applied for extension construction. Therefore, we suggest that a semantics-specific *fragment function* $\mathcal{FF}$ is applied to each autonomous fragment $F$ and returns a set of subsets of $F$. Each of these subsets is regarded as an elementary building block in extension construction. Moreover, if $\mathcal{AU}(S) = \{S\}$ the result should be the same as in the non *$\mathcal{AU}$-aware* case, therefore $\mathcal{FF}$ should be equal to $\mathcal{BF}$ in this case. More articulated considerations need to be applied when $|\mathcal{AU}(S)| \neq 1$ or $\mathcal{Y}(\mathcal{AU}(S)) \neq S$ (where $\mathcal{Y}(Q) \triangleq \bigcup_{P \in Q} P$, given that $Q$ is a set of sets). Let us consider orderly these cases.

If $|\mathcal{AU}(S)| \neq 1$, several autonomous fragments are considered separately and, for each autonomous fragment $F_i$, a set of subsets of $F_i$ is produced by $\mathcal{FF}$. They need then to be combined: the most direct way is considering all possible combinations of these subsets except those which infringe the conflict-free principle. In other words, all possible conflict-free combinations obtained by selecting one subset for each $F_i$ are considered.

If $\mathcal{Y}(\mathcal{AU}(S)) = S$, i.e. the whole $S$ has been considered since autonomous fragments are a partition of $S$, the above mentioned combinations represent the result of the application of $\mathcal{BF}'$ to $S$. Otherwise, there are some elements of $S$ which are not included in any autonomous fragment. We follow the idea that the inclusion in the extensions of the other elements of $S$ should be determined by the choices carried out within the autonomous fragments $\mathcal{AU}(S)$. In a sense, autonomous fragments are evaluated first, then the results of this evaluation are taken into account when the remaining elements of $S$ (if any) are considered.

In line with the fundamental principles of SCC-recursiveness, this amounts to invoke recursively the general function $\mathcal{GF}$ on a restricted argumentation framework, derived taking into account the choices in $\mathcal{AU}(S)$.

Having provided an outline of the underlying ideas, we need to put them in formal terms.

Let $\mathcal{BF}_{\mathcal{S}}$ be the base function of a SCC-recursive semantics $\mathcal{S}$; the corresponding *$\mathcal{AU}$-aware* base function $\mathcal{BF}'_{\mathcal{S}}$ is defined as follows.

**Definition 14** *Given a SCC-recursive semantics $\mathcal{S}$ with base function $\mathcal{BF}_{\mathcal{S}}(\mathrm{AF}, C)$, the corresponding $\mathcal{AU}$-*aware *base function $\mathcal{BF}'_{\mathcal{S}}(\mathrm{AF}, C)$ is a function that given an argumentation framework $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$ such that $|\mathrm{SCCS}_{\mathrm{AF}}| = 1$ (i.e. $\mathrm{SCCS}_{\mathrm{AF}} = \{\mathcal{A}\}$) and a set $C \subseteq \mathcal{A}$, gives a subset of $2^{\mathcal{A}}$ as follows:*
*$E \in \mathcal{BF}'_{\mathcal{S}}(\mathrm{AF}, C)$ if and only if $E$ is conflict-free and $(E \cap \mathcal{Y}(\mathcal{AU}(\mathcal{A}))) \in \mathcal{UCF}_{\mathcal{S}}(\mathcal{AU}(\mathcal{A}), \mathrm{AF}, C)$ and if $\mathcal{A} \setminus \mathcal{Y}(\mathcal{AU}(\mathcal{A})) \neq \emptyset$, $\forall S \in \mathrm{SCCS}_{\mathrm{AF}\downarrow_{\mathcal{A} \setminus \mathcal{Y}(\mathcal{AU}(\mathcal{A}))}}$ $(E \cap S) \in \mathcal{GF}'(\mathrm{AF}\downarrow_{UP_{\mathrm{AF}}(S,E)}, U_{\mathrm{AF}}(S, E) \cap C)$*

where

- $\mathcal{UCF}_{\mathcal{S}}(\Sigma, \mathrm{AF}, C)$ is a function which given a set $\Sigma$ of disjoint subsets of $\mathcal{A}$ returns a set of subsets of $\mathcal{Y}(\Sigma)$ as follows: $E \in \mathcal{UCF}_{\mathcal{S}}(\Sigma, \mathrm{AF}, C)$ if $E$ is conflict-free and $\forall Q \in \Sigma$, $E \cap Q = \mathcal{BF}_{\mathcal{S}}(\mathrm{AF}\downarrow_Q, C \cap Q)$;
- $\mathcal{GF}'$ is the general recursive function in the *$\mathcal{AU}$-aware* scheme (see Definition 15 below).

Definition 14 is rather complex and not really elegant. This reflects the preliminary state of this investigation: devising a simpler formulation is one of the directions of future work. To illustrate its main features we note that:

- the base function $\mathcal{BF}$ is applied to each autonomous fragment and the resulting sets are combined in all possible conflict-free manners (through function $\mathcal{UCF}$);
- the output of $\mathcal{UCF}$ is used directly as output of $\mathcal{BF}'$ if the union of all autonomous fragments $\mathcal{Y}(\mathcal{AU}(\mathcal{A}))$ completely covers the SCC $\mathcal{A}$, since in this case $\mathcal{A} \setminus \mathcal{Y}(\mathcal{AU}(\mathcal{A})) = \emptyset$ and therefore the second part of the definition does not apply;
- in particular, the *$\mathcal{AU}$-aware* base function $\mathcal{BF}'$ coincides with $\mathcal{BF}$ when there is only one autonomous fragment coinciding with the SCC $\mathcal{A}$ itself, since $\mathcal{UCF}_{\mathcal{S}}(\{\mathcal{A}\}, \mathrm{AF}, C)$ is invoked in this case, leading to $E \cap \mathcal{A} = \mathcal{BF}_{\mathcal{S}}(\mathrm{AF}{\downarrow_{\mathcal{A}}}, C \cap \mathcal{A}) = \mathcal{BF}_{\mathcal{S}}(\mathrm{AF}, C)$;
- if $\mathcal{A} \setminus \mathcal{Y}(\mathcal{AU}(\mathcal{A})) \neq \emptyset$, the construction of the output of $\mathcal{BF}'$ proceeds recursively using the output of $\mathcal{UCF}$ as starting point and invoking the general SCC-recursive function $\mathcal{GF}'$ "as usual" on the parts of $\mathcal{A}$ not covered by $\mathcal{Y}(\mathcal{AU}(\mathcal{A}))$, i.e. on the SCCs of the restricted argumentation framework $\mathrm{AF}{\downarrow_{\mathcal{A} \setminus \mathcal{Y}(\mathcal{AU}(\mathcal{A}))}}$.

Besides complications, taking into account the internal structure of SCCs has another downside: there are cases where $\mathcal{BF}'_{\mathcal{S}}(\mathrm{AF}, C) = \emptyset$. This may happen, for instance, when $\mathcal{A} \setminus \mathcal{Y}(\mathcal{AU}(\mathcal{A})) = \emptyset$ and $\mathcal{UCF}_{\mathcal{S}}(\mathcal{AU}(\mathcal{A}), \mathrm{AF}, C) = \emptyset$, i.e. the autonomous fragments cover the whole SCC and there are no conflict-free combinations of the subsets selected within them. This would lead to the unpleasant situation of non-existence of extensions for some argumentation frameworks. A (still not elegant) solution consists in introducing a provision for this case in the semantics definition.

**Definition 15** *Given a SCC-recursive semantics $\mathcal{S}$ with base function $\mathcal{BF}_{\mathcal{S}}(\mathrm{AF}, C)$, the corresponding $\mathcal{AU}$-*aware *semantics $\mathcal{S}'$ is defined as follows: for any argumentation framework $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$, $\mathcal{E}_{\mathcal{S}'}(\mathrm{AF}) = \mathcal{GF}'(\mathrm{AF}, \mathcal{A})$, where for any $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$ and for any set $C \subseteq \mathcal{A}$, the function $\mathcal{GF}'(\mathrm{AF}, C) \subseteq 2^{\mathcal{A}}$ is defined as follows:*
*for any $E \subseteq \mathcal{A}$, $E \in \mathcal{GF}'(\mathrm{AF}, C)$ if and only if*

- *in case $|\mathrm{SCCS_{AF}}| = 1$, $E \in \mathcal{BF}^*_{\mathcal{S}}(\mathrm{AF}, C)$*
- *otherwise, $\forall S \in \mathrm{SCCS_{AF}}$ $(E \cap S) \in \mathcal{GF}'(\mathrm{AF}{\downarrow_{UP_{\mathrm{AF}}(S,E)}}, U_{\mathrm{AF}}(S, E) \cap C)$*

where $\mathcal{BF}^*_{\mathcal{S}}(\mathrm{AF}, C) = \mathcal{BF}'_{\mathcal{S}}(\mathrm{AF}, C)$ if $\mathcal{BF}'_{\mathcal{S}}(\mathrm{AF}, C) \neq \emptyset$, $\mathcal{BF}^*_{\mathcal{S}}(\mathrm{AF}, C) = \{\emptyset\}$ otherwise.

Essentially an *$\mathcal{AU}$-aware* semantics is just a SCC-recursive semantics with a special base function $\mathcal{BF}^*$, which, apart a particular case, coincides with the *$\mathcal{AU}$-aware* base function $\mathcal{BF}'$. In turn, $\mathcal{BF}'$ exploits the original base function $\mathcal{BF}$ in the cases where a SCC does not admit significant autonomous fragments.

Proving fundamental properties of *$\mathcal{AU}$-aware* semantics turns out to be relatively easy, in virtue of its adherence to the general SCC-recursive scheme, whose properties are analyzed in (Baroni, Giacomin, & Guida 2005).

Let us consider well-foundedness of recursion first.

**Proposition 4** *Recursion in Definition 14 is well-founded.*

Definition 14 involves an indirect recursion: it invokes the $\mathcal{GF}'$ function on each SCC $S$ of $\mathrm{AF}{\downarrow_{\mathcal{A} \setminus \mathcal{Y}(\mathcal{AU}(\mathcal{A}))}}$ and such function in turn invokes the *$\mathcal{AU}$-aware* base function $\mathcal{BF}'_{\mathcal{S}}$ in the first branch of Definition 15. To verify the well-foundedness of this indirect recursion, first note that since $\mathcal{AU}(\mathcal{A}) \neq \emptyset$ (Proposition 3) and, by Definition 12, $\forall F \in \mathcal{AU}(\mathcal{A})$ $F \neq \emptyset$, it turns out that $\mathcal{Y}(\mathcal{AU}(\mathcal{A})) \neq \emptyset$. As a consequence, the restricted argumentation framework considered in the recursive branch $\mathrm{AF}{\downarrow_{\mathcal{A} \setminus \mathcal{Y}(\mathcal{AU}(\mathcal{A}))}}$ (and therefore also any of its SCCs) has a strictly lesser number of arguments than $|\mathcal{A}|$. Observe also that if $\mathcal{A} \setminus \mathcal{Y}(\mathcal{AU}(\mathcal{A})) = \emptyset$, the recursive part of Definition 14 is not invoked. This implies that subsequent invocations (if any) of the recursive branch of $\mathcal{BF}'_{\mathcal{S}}$ (reached through $\mathcal{GF}'$) operate on progressively smaller non-empty argumentation frameworks. Due to the hypothesis of finiteness of $\mathcal{A}$, this leads to consider the case where $\mathcal{BF}'_{\mathcal{S}}$ is invoked on an argumentation framework $\mathrm{AF} = \langle \mathcal{A}, \rightarrow \rangle$ such that $|\mathcal{A}| = 1$. Such argumentation framework consists necessarily of a unique SCC: by Definition 13, in this case it holds $\mathcal{AU}(\mathcal{A}) = \{\mathcal{A}\}$, and, as a consequence, $\mathcal{A} \setminus \mathcal{Y}(\mathcal{AU}(\mathcal{A})) = \emptyset$ which represents the non-recursive case of Definition 14. □

**Proposition 5** *The definition of an $\mathcal{AU}$-*aware *SCC-recursive semantics $\mathcal{S}'$ is well-founded.*

The definition of an *$\mathcal{AU}$-aware* SCC-recursive semantics is a case of the general SCC-recursive scheme with a well-defined base function (Proposition 4). Then the conclusion directly derives from the properties of the general SCC-recursive scheme shown in (Baroni, Giacomin, & Guida 2005). □

In the same line, we now show that an *$\mathcal{AU}$-aware* SCC-recursive semantics $\mathcal{S}'$ shares with its non *$\mathcal{AU}$-aware* version the fundamental properties of being conflict-free and agreeing with grounded semantics.

**Proposition 6** *Any $\mathcal{AU}$-*aware *SCC-recursive semantics $\mathcal{S}'$ satisfies the conflict-free property.*

As recalled in Proposition 1, for any SCC-recursive semantics $\mathcal{T}$, if its base function $\mathcal{BF}_{\mathcal{T}}$ is conflict-free then $\mathcal{T}$ satisfies the conflict-free property. Since $\mathcal{S}'$ is a special case of SCC-recursive semantics, where $\mathcal{BF}^*_{\mathcal{S}}$ plays the role of $\mathcal{BF}_{\mathcal{T}}$, it is sufficient to show that $\mathcal{BF}^*_{\mathcal{S}}$ is conflict free, namely that all elements of $\mathcal{BF}^*_{\mathcal{S}}(\mathrm{AF}, C)$ are conflict free. By Definition 15, this in turn corresponds to require that $\mathcal{BF}'_{\mathcal{S}}(\mathrm{AF}, C)$ is conflict-free, which holds by Definition 14. □

As to agreement with grounded semantics, it has to be verified that if the sufficient condition for agreement stated in Proposition 2 is satisfied by $\mathcal{BF}$, then it is satisfied also by $\mathcal{BF}^*$.

**Proposition 7** *If a SCC-recursive semantics $\mathcal{S}$ satisfies the hypothesis of Proposition 2, the corresponding $\mathcal{AU}$-*aware *semantics $\mathcal{S}'$ satisfies it as well.*

We need to show that $\mathcal{BF}^*_{\mathcal{S}}(\langle \{\alpha\}, \emptyset \rangle, \{\alpha\}) = \{\{\alpha\}\}$. This immediately follows from the fact that $\langle \{\alpha\}, \emptyset \rangle$

is an argumentation framework consisting of a single monadic SCC and therefore $\mathcal{BF}^*_{\mathcal{S}}(\langle\{\alpha\},\emptyset\rangle,\{\alpha\}) = \mathcal{BF}'_{\mathcal{S}}(\langle\{\alpha\},\emptyset\rangle,\{\alpha\}) = \mathcal{BF}_{\mathcal{S}}(\langle\{\alpha\},\emptyset\rangle,\{\alpha\}) = \{\{\alpha\}\}$. □

## Putting $\mathcal{AU}$-*aware* semantics at work

Having shown that, despite its rather articulated form, our attempt to define $\mathcal{AU}$-*aware* semantics preserves the fundamental properties that are desirable for any semantics, its practical impact remains to be analyzed. First, let us remark that in all the examples discussed in (Baroni, Giacomin, & Guida 2005) every SCC $S$ is such that $\mathcal{AU}(S) = \{S\}$ and therefore no differences emerge when considering an $\mathcal{AU}$-*aware* semantics wrt. its non $\mathcal{AU}$-*aware* version. Let us now review the motivating examples introduced above, examining the behavior of the $\mathcal{AU}$-*aware* versions of preferred and $CF2$ semantics, denoted as $\mathcal{PR}'$ and $CF2'$ respectively.

Since all examples involve an argumentation framework $\mathrm{AF}_i$ consisting of a single SCC $S_i$, Definition 15 directly leads to consider the following invocation of the $\mathcal{AU}$-*aware* base function: $\mathcal{E}_{\mathcal{S}'}(\mathrm{AF}_i) = \mathcal{BF}'_{\mathcal{S}}(\mathrm{AF}_i, S_i)$.

**Example 1.** Let us start with $\mathrm{AF}_1$ (Figure 1), recalling that $\mathcal{AU}(S_1) = \{\{\alpha\}\}$ and, therefore, $\mathcal{Y}(\mathcal{AU}(S_1)) = \{\alpha\}$. Since $\mathcal{Y}(\mathcal{AU}(S_1)) \subsetneq S_1$ both parts of Definition 14 apply. The first part states that for any extension $E$, $E \cap \mathcal{Y}(\mathcal{AU}(S_1)) = \mathcal{UCF}_{\mathcal{S}}(\mathcal{AU}(S_1), \mathrm{AF}_1, S_1)$. Since $\mathcal{AU}(S_1)$ contains just one element, $\mathcal{UCF}_{\mathcal{S}}(\mathcal{AU}(S_1), \mathrm{AF}_1, S_1) = \mathcal{BF}_{\mathcal{S}}(\mathrm{AF}_1{\downarrow_{\{\alpha\}}}, \{\alpha\})$. Since $\mathrm{AF}_1{\downarrow_{\{\alpha\}}}$ consists of a single self-defeating argument and $\mathcal{BF}_{\mathcal{S}}(\mathrm{AF}_1{\downarrow_{\{\alpha\}}}, \{\alpha\}) = \{\emptyset\}$ either with $\mathcal{S} = CF2$ or $\mathcal{S} = \mathcal{PR}$, it turns out that $E \cap \{\alpha\} = \emptyset$. In words, $\alpha$ can not be included in any extension.

Then, according to the second part of Definition 14, $E \cap T$ is computed recursively for all $T \in \mathrm{SCCS}_{\mathrm{AF}_1\downarrow_{S_1\setminus\{\alpha\}}} = \{\{\beta\},\{\gamma\}\}$. Following the SCC order within $\mathrm{AF}_1{\downarrow_{S_1\setminus\{\alpha\}}}$, $\{\beta\}$ has to be considered first, yielding $E \cap \{\beta\} = \mathcal{GF}'(\mathrm{AF}_1{\downarrow_{UP_{\mathrm{AF}_1}(\{\beta\},E)}}, U_{\mathrm{AF}_1}(\{\beta\},E) \cap S_1)$.Since $E \cap \mathcal{Y}(\mathcal{AU}(S_1)) = E \cap \{\alpha\} = \emptyset$, it turns out that $\beta$ is not attacked by $E$ nor is defended by $E$ from the attack coming from $\alpha$, therefore $UP_{\mathrm{AF}_1}(\{\beta\},E) = \{\beta\}$ and $U_{\mathrm{AF}_1}(\{\beta\},E) = \emptyset$. As a consequence, $E \cap \{\beta\} = \mathcal{GF}'(\mathrm{AF}_1{\downarrow_{\{\beta\}}}, \emptyset)$, which by the first clause of Definition 14 yields $E \cap \{\beta\} = \mathcal{BF}'_{\mathcal{S}'}(\langle\{\beta\},\emptyset\rangle,\emptyset)$, resulting in $E \cap \{\beta\} = \{\beta\}$ with $\mathcal{S} = CF2$, and in $E \cap \{\beta\} = \emptyset$ with $\mathcal{S} = \mathcal{PR}$.

Turning to $\{\gamma\}$, we have $E \cap \{\gamma\} \in \mathcal{GF}'(\mathrm{AF}_1{\downarrow_{UP_{\mathrm{AF}_1}(\{\gamma\},E)}}, U_{\mathrm{AF}_1}(\{\gamma\},E) \cap S_1)$. In the case of $CF2$ semantics, since $E \cap \{\beta\} = \{\beta\}$ for any $E$, we have $UP_{\mathrm{AF}_1}(\{\gamma\},E) = U_{\mathrm{AF}_1}(\{\gamma\},E) = \emptyset$, which skipping some further purely formal steps leads to consider the empty argumentation framework and therefore to conclude $E \cap \{\gamma\} = \emptyset$. In the case of preferred semantics, it holds that $UP_{\mathrm{AF}_1}(\{\gamma\},E) = \{\gamma\}$, $U_{\mathrm{AF}_1}(\{\gamma\},E) = \emptyset$, which (skipping again some steps) gives $E \cap \{\gamma\} = \emptyset$.

Summing up, we obtain $\mathcal{E}_{CF2'}(\mathrm{AF}_1) = \{\{\beta\}\}$, while $\mathcal{E}_{\mathcal{PR}'}(\mathrm{AF}_1) = \emptyset$. This shows that the $\mathcal{AU}$-*aware* version of $CF2$ semantics provides a different (and intuitively more acceptable) result wrt. the non $\mathcal{AU}$-*aware* one.

**Example 2.** In the case of $\mathrm{AF}_2$ (Figure 2), $\mathcal{AU}(S_2) = \{\{\alpha,\beta,\gamma\}\}$. As to the first part of Definition 14, for any extension $E$, $E \cap \mathcal{Y}(\mathcal{AU}(S_2)) = \mathcal{UCF}_{\mathcal{S}}(\mathcal{AU}(S_2), \mathrm{AF}_2, S_2) = \mathcal{BF}_{\mathcal{S}}(\mathrm{AF}_2{\downarrow_{\{\alpha,\beta,\gamma\}}}, \{\alpha,\beta,\gamma\})$. Here the two semantics differ since in the case of preferred semantics $E \cap \{\alpha,\beta,\gamma\} \in \mathcal{BF}_{\mathcal{PR}}(\mathrm{AF}_2{\downarrow_{\{\alpha,\beta,\gamma\}}}, \{\alpha,\beta,\gamma\}) = \{\emptyset\}$, while in the case of $CF2$-semantics $E \cap \{\alpha,\beta,\gamma\} \in \mathcal{BF}_{CF2}(\mathrm{AF}_2{\downarrow_{\{\alpha,\beta,\gamma\}}}, \{\alpha,\beta,\gamma\}) = \{\{\alpha\},\{\beta\},\{\gamma\}\}$. In both cases, $E \cap T$ has to be computed recursively for all $T \in \mathrm{SCCS}_{\mathrm{AF}_2\downarrow_{S_2\setminus\{\alpha,\beta,\gamma\}}} = \{\{\delta\},\{\epsilon\}\}$, on the basis of the choices carried out for $E \cap \{\alpha,\beta,\gamma\}$.

Let us examine the case of preferred semantics first, where there is just one choice for $E \cap \{\alpha,\beta,\gamma\} = \emptyset$. Following the SCC order within $\mathrm{AF}_2{\downarrow_{S_2\setminus\{\alpha,\beta,\gamma\}}}$, $\{\delta\}$ has to be considered first. Since $\{\delta\}$ is exactly in the same situation as $\{\beta\}$ in Example 1, skipping the analogous formal steps made explicit in Example 1 we obtain $E \cap \{\delta\} = \emptyset$. Consequently, when considering the subsequent SCC of $\mathrm{AF}_2{\downarrow_{S_2\setminus\{\alpha,\beta,\gamma\}}}$, namely $\{\epsilon\}$, we are in a completely analogous situation as for $\{\gamma\}$ in Example 1 and we obtain $E \cap \{\epsilon\} = \emptyset$.

In summary, we obtain $\mathcal{E}_{\mathcal{PR}'}(\mathrm{AF}_2) = \mathcal{E}_{\mathcal{PR}}(\mathrm{AF}_2) = \{\emptyset\}$.

Let us turn to $CF2$-semantics, where there are three choices for $E \cap \{\alpha,\beta,\gamma\}$, namely $\{\alpha\}$, $\{\beta\}$, and $\{\gamma\}$, each being the starting point for the construction of one or more extensions, to be completed by possibly adding elements of $\mathrm{AF}_2{\downarrow_{S_2\setminus\{\alpha,\beta,\gamma\}}}$.

Consider first the case where $E \cap \{\alpha,\beta,\gamma\} = \{\alpha\}$. We note that in this case $E$ defends $\delta$ by attacking $\beta$, and therefore $UP_{\mathrm{AF}_2}(\{\delta\},E) = U_{\mathrm{AF}_2}(\{\delta\},E) = \{\delta\}$. Thus $\delta$ and $\epsilon$ are in the same situation as $\beta$ and $\gamma$ respectively in $\mathrm{AF}_1$. This leads to $E \cap \{\delta\} = \{\delta\}$ and $E \cap \{\epsilon\} = \emptyset$, obtaining a first extension $E_1 = \{\alpha,\delta\}$.

Let us now examine the case $E \cap \{\alpha,\beta,\gamma\} = \{\beta\}$. In this case $E$ attacks $\delta$, and therefore $UP_{\mathrm{AF}_2}(\{\delta\},E) = U_{\mathrm{AF}_2}(\{\delta\},E) = \emptyset$; skipping some purely formal steps this clearly leads to $E \cap \{\delta\} = \emptyset$. As a consequence, it turns out that $UP_{\mathrm{AF}_2}(\{\epsilon\},E) = U_{\mathrm{AF}_2}(\{\epsilon\},E) = \{\epsilon\}$, which leads to $E \cap \{\epsilon\} = \{\epsilon\}$, thus obtaining a second extension $E_2 = \{\beta,\epsilon\}$.

Finally, assume $E \cap \{\alpha,\beta,\gamma\} = \{\gamma\}$. In this case $E$ neither attacks nor defends $\delta$, therefore $U_{\mathrm{AF}_2}(\{\delta\},E) = \emptyset$, while $UP_{\mathrm{AF}_2}(\{\delta\},E) = \{\delta\}$. Since only $UP_{\mathrm{AF}_2}(\{\delta\},E)$ is relevant in the definition of $\mathcal{BF}_{CF2}$, this gives rise to $E \cap \{\delta\} = \{\delta\}$ and, consequently, $E \cap \{\epsilon\} = \emptyset$, obtaining a third extension $E_3 = \{\gamma,\delta\}$.

Summing up, $\mathcal{E}_{CF2'}(\mathrm{AF}_2) = \{\{\alpha,\delta\},\{\beta,\epsilon\},\{\gamma,\delta\}\} \neq \mathcal{E}_{CF2}(\mathrm{AF}_2) = \{\{\alpha,\delta\},\{\alpha,\epsilon\},\{\beta,\epsilon\},\{\gamma,\delta\}\}$. As to the justification status of arguments, the difference does not manifest itself, since, according to the $\mathcal{AU}$-*aware* version of $CF2$ semantics too, no argument is included in all extensions. It is however interesting that the extension $\{\alpha,\epsilon\}$ is not prescribed by the $\mathcal{AU}$-*aware* version of $CF2$ semantics, as not compatible with the idea of choosing first within the autonomous fragment $\{\alpha,\beta,\gamma\}$ and then propagating the effects on the rest of the argumentation framework.

Neither the $\mathcal{AU}$-*aware* version of preferred semantics nor of $CF2$ captures the intuition underlying the example that the three-length cycle $\{\alpha,\beta,\gamma\}$ could be regarded as a sort of "null element", leaving $\delta$ undefeated and $\gamma$ defeated. The search for a semantics featuring this kind of behavior remains open.

**Example 3.** Consider now $\mathrm{AF}_3$ (Figure 3). Since $\mathcal{AU}(S_3) = \{S_3\}$ the behavior of any non *$\mathcal{AU}$-aware* semantics and of its *$\mathcal{AU}$-aware* version is the same and therefore will not be discussed here.

**Example 4.** In $\mathrm{AF}_4$ (Figure 4), $\mathcal{AU}(S_4) = \{\{\alpha, \beta\}, \{\gamma, \delta\}\}$. In this case $\mathcal{Y}(\mathcal{AU}(S_4)) = S_4$, and therefore $\mathcal{E}_{\mathcal{S}'}(\mathrm{AF}_4) = \mathcal{UCF}_{\mathcal{S}}(\mathcal{AU}(S_4), \mathrm{AF}_4, S_4)$. This means that first the base function $\mathcal{BF}_{\mathcal{S}}$ is evaluated separately for $\mathrm{AF}_4{\downarrow_{\{\alpha,\beta\}}}$ and $\mathrm{AF}_4{\downarrow_{\{\gamma,\delta\}}}$. Both fragments consist of a couple of rebutting arguments, a prototypical case often referred to as "Nixon diamond" where any multiple-status semantics admits two extensions, each corresponding to the choice of one of the arguments. We have therefore $\mathcal{BF}_{\mathcal{S}}(\mathrm{AF}_4{\downarrow_{\{\alpha,\beta\}}}, S_4 \cap \{\alpha, \beta\}) = \{\{\alpha\}, \{\beta\}\}$, and $\mathcal{BF}_{\mathcal{S}}(\mathrm{AF}_4{\downarrow_{\{\gamma,\delta\}}}, S_4 \cap \{\gamma, \delta\}) = \{\{\gamma\}, \{\delta\}\}$ both with $\mathcal{S} = \mathcal{PR}$ and $\mathcal{S} = CF2$. All conflict free combinations of the elements of $\{\{\alpha\}, \{\beta\}\}$ and $\{\{\gamma\}, \{\delta\}\}$ are then returned by function $\mathcal{UCF}$, yielding $\mathcal{E}_{\mathcal{S}'}(\mathrm{AF}_4) = \{\{\alpha, \delta\}, \{\beta, \gamma\}\} = \mathcal{E}_{\mathcal{S}}(\mathrm{AF}_4)$ both with $\mathcal{S} = \mathcal{PR}$ and $\mathcal{S} = CF2$. Therefore, in this case the use of *$\mathcal{AU}$-aware* semantics does not give rise to different results.

**Example 5.** Differences appear instead in $\mathrm{AF}_5$ (Figure 5), where $\mathcal{AU}(S_5) = \{\{\alpha, \beta\}, \{\delta\}\}$.

First, $\mathcal{UCF}_{\mathcal{S}}(\mathcal{AU}(S_5), \mathrm{AF}_5, S_5)$ has to be evaluated, which requires in turn evaluating $\mathcal{BF}_{\mathcal{S}}(\mathrm{AF}_5{\downarrow_Q}, S_5 \cap Q)$ for $Q \in \{\{\alpha, \beta\}, \{\delta\}\}$. We have $\mathcal{BF}_{\mathcal{S}}(\mathrm{AF}_5{\downarrow_{\{\alpha,\beta\}}}, S_5 \cap \{\alpha, \beta\}) = \{\{\alpha\}, \{\beta\}\}$, and $\mathcal{BF}_{\mathcal{S}}(\mathrm{AF}_5{\downarrow_{\{\delta\}}}, S_5 \cap \{\delta\}) = \{\emptyset\}$ both with $\mathcal{S} = \mathcal{PR}$ and $\mathcal{S} = CF2$.

Considering the conflict free combinations, we derive $\mathcal{UCF}_{\mathcal{S}}(\mathcal{AU}(S_5), \mathrm{AF}_5, S_5) = \{\{\alpha\}, \{\beta\}\}$.

Therefore, with both semantics, we have two starting choices for $E \cap \mathcal{Y}(\mathcal{AU}(S_5))$, namely $\{\alpha\}$ and $\{\beta\}$. The restricted argumentation framework $\mathrm{AF}_5{\downarrow_{\{\gamma\}}}$ remains to be considered, which clearly consists of a single SCC $\{\gamma\}$. Consider first the case $E \cap \mathcal{Y}(\mathcal{AU}(S_5)) = \{\alpha\}$; since $E$ does not attack $\gamma$ and $\gamma$ defends itself against the attack coming from $\delta$, we have $UP_{\mathrm{AF}_5}(\{\gamma\}, E) = U_{\mathrm{AF}_5}(\{\gamma\}, E) = \{\gamma\}$, and therefore we obtain $\mathcal{BF}_{\mathcal{S}}(\mathrm{AF}_5{\downarrow_{\{\gamma\}}}, S_5 \cap \{\gamma\}) = \{\gamma\}$, both with $\mathcal{S} = \mathcal{PR}$ and $\mathcal{S} = CF2$. This leads to consider $E = \{\alpha, \gamma\}$ which, not being conflict free, is not compatible with Definition 14 and is discarded.

The development of the case $E \cap \mathcal{Y}(\mathcal{AU}(S_5)) = \{\beta\}$ is analogous and leads to consider $E = \{\beta, \gamma\}$ which is conflict free and compatible with Definition 14.

In summary, $\mathcal{E}_{\mathcal{S}'}(\mathrm{AF}_5) = \{\{\beta, \gamma\}\}$ both with $\mathcal{S} = \mathcal{PR}$ and $\mathcal{S} = CF2$. Note that this is the same result as for non *$\mathcal{AU}$-aware* preferred semantics, since $\mathcal{E}_{\mathcal{PR}}(\mathrm{AF}_5) = \{\{\beta, \gamma\}\}$ while a difference appears for $CF2$ semantics where $\mathcal{E}_{CF2}(\mathrm{AF}_5) = \{\{\beta, \gamma\}, \{\alpha\}\}$.

Thus the *$\mathcal{AU}$-aware* version of $CF2$ semantics agrees with preferred semantics in this case (while the non *$\mathcal{AU}$-aware* version does not) and achieves a behavior which is intuitively plausible if self-defeating arguments are considered as "null elements" in an argumentation framework.

## Conclusions

We have provided an initial investigation about the potential use of the novel notion of autonomous fragments within SCC-recursive argumentation semantics. The presented results are quite preliminary and further work is needed in order to improve the definition of *$\mathcal{AU}$-aware* semantics and explore more deeply its properties. Though our analysis has started from specific examples, we remark that the aim of the paper is not to achieve a "better" treatment of particular cases but rather to suggest an interesting perspective about argumentation semantics design. In fact, it emerges that different solutions are obtained by changing the semantics behavior with respect to topology, e.g. choosing between the *$\mathcal{AU}$-aware* and the non *$\mathcal{AU}$-aware* version of a semantics, without affecting the underlying notion of extension, represented by the base function $\mathcal{BF}$. This suggest that the "design" of an argumentation semantics can be conceived as composed, in a modular way, by the answers to two "orthogonal" questions: i) how to take into account the defeat graph topology in extension construction and ii) which principles rule the identification of extensions within the basic topological entities considered. Identifying alternative answers to these questions and properly combining and comparing them appears to be a very interesting research line to pursue.

## References


Baroni, P., and Giacomin, M. 2003. Solving semantic problems with odd-length cycles in argumentation. In *Proc. of the 7th European Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2003)*, 440–451.

Baroni, P., and Giacomin, M. 2004a. A general recursive schema for argumentation semantics. In *Proc. of the 16th European Conf. on Artificial Intelligence (ECAI 2004)*, 783–787.

Baroni, P., and Giacomin, M. 2004b. A recursive approach to argumentation: motivation and perspectives. In *Proc. of the 10th Int. Workshop on Non-Monotonic Reasoning (NMR 2004)*, 50–58.

Baroni, P.; Giacomin, M.; and Guida, G. 2005. Scc-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence* 168(1-2):165–210.

Caminada, M., and Amgoud, L. 2005. An axiomatic account of formal argumentation. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005)*, 608–613.

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.

Horty, J. F. 2002. Skepticism and floating conclusions. *Artificial Intelligence* 135(1–2):55–72.

Pollock, J. L. 1992. How to reason defeasibly. *Artificial Intelligence* 57(1):1–42.

Prakken, H., and Vreeswijk, G. A. W. 2001. Logics for defeasible argumentation. In Gabbay, D. M., and Guenthner, F., eds., *Handbook of Philosophical Logic, Second Edition*. Dordrecht: Kluwer Academic Publishers.