dbai

TECHNICAL
REPORT

INSTITUT FÜR INFORMATIONSSYSTEME

ABTEILUNG DATENBANKEN UND ARTIFICIAL INTELLIGENCE

The cf2 Argumentation Semantics Revisited

DBAI-TR-2012-77

Sarah Alice Gaggl   **Stefan Woltran**

DBAI TECHNICAL REPORT

2012

Institut für Informationssysteme

Abteilung Datenbanken und

Artificial Intelligence

Technische Universität Wien

Favoritenstr. 9

A-1040 Vienna, Austria

Tel: +43-1-58801-18403

Fax: +43-1-58801-18493

sekret@dbai.tuwien.ac.at

www.dbai.tuwien.ac.at

TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

The cf2 Argumentation Semantics Revisited

Sarah Alice Gaggl [1]   Stefan Woltran [1]

Abstract. Abstract argumentation frameworks nowadays provide the most popular formalization of argumentation on a conceptual level. Numerous semantics for this paradigm have been proposed, whereby the *cf2* semantics has shown to solve particular problems concerned with odd-length cycles in such frameworks. Due to the complicated definition of this semantics it has somehow been neglected in the literature. In this article, we introduce an alternative characterization of the *cf2* semantics which, roughly speaking, avoids the recursive computation of sub-frameworks. This facilitates further investigation steps, like a complete complexity analysis. Furthermore, we show how the notion of strong equivalence can be characterized in terms of the *cf2* semantics. In contrast to other semantics, it turns out that for the *cf2* semantics strong equivalence coincides with syntactical equivalence. We make this particular behavior more explicit by defining a new property for argumentation semantics, called the succinctness property. If a semantics σ satisfies the succinctness property, then for every framework F, all its attacks contribute to the evaluation of at least one framework F' containing F. We finally characterize strong equivalence also for the stage and the naive semantics. Together with known results these characterizations imply that none of the prominent semantics for abstract argumentation, except the *cf2* semantics, satisfies the succinctness property.

[1]Institute of Information Systems 184/2, Technische Universität Wien, Favoritenstrasse 9-11, 1040 Vienna, Austria. E-mail: gaggl,woltran@dbai.tuwien.ac.at

1 Introduction

Abstract argumentation frameworks (AFs), introduced by Dung [11], represent the most popular approach for formalizing and reasoning over argumentation problems on a conceptual level. Dung already introduced different extension-based semantics (preferred, complete, stable, grounded) for such frameworks. In addition, recent proposals tried to overcome several shortcomings observed for those original semantics. For instance, the semi-stable semantics [8], and likewise the stage semantics [23], handles the problem of the possible non-existence of stable extensions, while the ideal semantics [12] is proposed as a unique-status approach (each AF possesses exactly one extension) less skeptical than the grounded extension.

Another family of semantics, the so-called SCC-recursive semantics has been introduced in [7]. Hereby, a recursive decomposition of the given AF along strongly connected components (SCCs) is necessary to obtain the extensions. Among them, the *cf2* semantics, first proposed in [3] and later discussed in [7], has been introduced in order to solve particular problems arising for AFs with odd-length cycles. It fulfills several requirements such as the symmetric treatment of odd- and even-length cycles, and ensures that attacks from self-defeating arguments have no influence on the selection of other arguments to be included in an extension. Furthermore, the *cf2* semantics satisfies most of the evaluation criteria proposed in [4]. Basically, only the admissibility- and reinstatement criteria are violated. This is due to fact that *cf2* semantics explicitly gives up on these conditions when it comes to evaluate SCCs. At this point, it has to be mentioned however that the property of admissibility turns out to be crucial when abstract argumentation is employed in certain forms of instantiation-based argumentation, see e.g. [9].

Due to the quite complicated definition of its semantics, the *cf2* approach has been somehow neglected in the literature. For instance, a complete complexity analysis is still missing, although Nieves et al. observed in [19] that the decision problem of verifying whether a set of arguments is a *cf2* extension is polynomial-time computable. However, *cf2* semantics attracted specific attention lately, for example in [1] it has been used to handle loops in Talmudic Logic.

In another branch of research, attention was directed to the investigation of redundant patterns in AFs. Oikarinen and Woltran [20] identified kernels which eliminate those redundant attacks of AFs and introduced the concept of strong equivalence: Two AFs are strongly equivalent wrt. a semantics σ (i.e. they provide the same σ-extensions no matter how the two AFs are simultaneously extended), if their σ-kernels coincide. In [2] the notion of equivalence wrt. stable semantics has been studied also for logic-based argumentation systems. To the best of our knowledge, redundancies for the *cf2* semantics have not been studied yet. As we show in this paper, the *cf2* semantics have the interesting feature that strong equivalence coincides with syntactical equivalence. In other words, for the *cf2* semantics there are no redundant attacks.

The main contributions of this article are the following.

- To simplify further investigation, we first give an alternative characterization for the *cf2* semantics. The original definition of Baroni et al. [7] involves a recursive computation of different sub-frameworks. Our aim here is to shift the need of recursion from generating sub-frameworks to arguments. We show that the required set of arguments can be captured via a fixed-point operator. This allows to characterize *cf2* semantics using only linear recursion.

- With the alternative characterization at hand, we formally prove the following complexity results. (1) Verifying if a given set is a $cf2$ extension is in P; (2) deciding if an argument is contained in some $cf2$ extension (credulous acceptance) is NP-complete; (3) deciding if an argument is contained in all $cf2$ extension (skeptical acceptance) is coNP-complete; and (4) checking whether there exists a nonempty $cf2$ extension is in P.

- As the third main contribution we define a new property for argumentation semantics called the *succinctness property*. As outlined above, a semantics satisfies the succinctness property or is *maximal succinct* iff no redundant attacks for this semantics do exist. It turns out that the $cf2$ semantics is the only one which is maximal succinct, whereas for other semantics we can reuse results about strong equivalence [20] for an analysis on their succinctness. Our results thus provide a new classification for argumentation semantics, namely in terms of redundant attacks.

Parts of this article have been published in proceedings of conferences [17, 18]. Completely novel material is provided by the complexity analysis as well as by the investigations on the succinctness property.

2 Preliminaries

In this section we introduce the basics of abstract argumentation, the semantics we need for further investigations and some properties of the semantics we are mainly interested in this work, the $cf2$ semantics.

2.1 Abstract Argumentation

The definition of abstract argumentation frameworks and the semantics are based on [11, 23].

Definition 2.1 *An **argumentation framework** (AF) is a pair $F = (A, R)$, where A is a finite set of arguments and $R \subseteq A \times A$. The pair $(a, b) \in R$ means that a attacks b. A set $S \subseteq A$ of arguments **defeats** b (in F), if there is an $a \in S$, such that $(a, b) \in R$. An argument $a \in A$ is **defended** by $S \subseteq A$ (in F) iff, for each $b \in A$, it holds that, if $(b, a) \in R$, then S defeats b (in F).*

The inherent conflicts between the arguments are solved by selecting subsets of arguments, where a semantics σ assigns a collection of sets of arguments to an AF F. The basic requirement for all semantics is that none of the selected arguments attack each other.

Definition 2.2 *Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is said to be **conflict-free** (in F), if there are no $a, b \in S$, such that $(a, b) \in R$. We denote the collection of sets which are conflict-free (in F) by $cf(F)$. A set $S \subseteq A$ is maximal conflict-free or **naive**, if $S \in cf(F)$ and for each $T \in cf(F)$, $S \not\subset T$. We denote the collection of all naive sets of F by $naive(F)$. For the empty AF $F_0 = (\emptyset, \emptyset)$, we set $naive(F_0) = \{\emptyset\}$.*

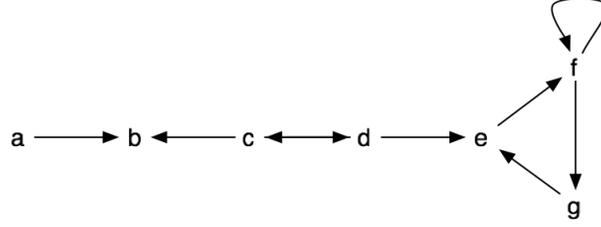Beside the naive semantics we will consider the following semantics in this work.

Figure 1: The argumentation framework F from Example 2.4.

Definition 2.3 *Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is said to be*

- *a **stable** extension (of F), i.e. $S \in stable(F)$, if $S \in cf(F)$ and each $a \in A \setminus S$ is defeated by S in F.*

- *a **stage** extension (of F), i.e. $S \in stage(F)$, if $S \in cf(F)$ and there is no $T \in cf(F)$ with $T_R^+ \supset S_R^+$, where $S_R^+ = S \cup \{b \mid \exists a \in S, s.\ t.\ (a, b) \in R\}$.*

- *an **admissible** extension, i.e. $S \in adm(F)$ if $S \in cf(F)$ and each $a \in S$ is defended by S.*

- *a **preferred** extension, i.e. $S \in pref(F)$ if $S \in adm(F)$ and for each $T \in adm(F)$, $S \not\subset T$.*

We illustrate the different behavior of the introduced semantics in the following example.

Example 2.4 *Consider the AF $F = (A, R)$ with $A = \{a, b, c, d, e, f, g\}$ and $R = \{(a, b), (c, b), (c, d), (d, c), (d, e), (e, f), (f, f), (f, g), (g, e)\}$ as in Figure 1. Then, the above defined semantics yield the following extensions.*

- $stable(F) = \emptyset$;

- $naive(F) = stage(F) = \{\{a, d, g\}, \{a, c, e\}, \{a, c, g\}\}$;

- $adm(F) = \{\{\}, \{a\}, \{c\}, \{d\}, \{a, c\}, \{a, d\}\}$;

- $pref(F) = \{\{a, c\}, \{a, d\}\}$.

$\diamond$

2.2 The *cf2* Semantics

The semantics we are mainly interested in this work is based on a decomposition along the strongly connected components (SCCs) of an AF. Hence, we require some further formal machinery. A directed graph is called *strongly connected* if there is a path from each vertex in the graph to every other vertex of the graph. By $SCCs(F)$, we denote the set of *strongly connected components* of an AF $F = (A, R)$, i.e. sets of vertices of the maximal strongly connected sub-graphs of F; $SCCs(F)$
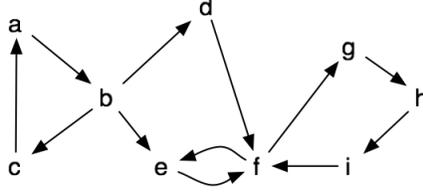
Figure 2: The argumentation framework F from Example 2.5.

is thus a partition of A. Moreover, for an argument $a \in A$, we denote by $C_F(a)$ the component of F where a occurs in, i.e. the (unique) set $C \in SCCs(F)$, such that $a \in C$. AFs $F_1 = (A_1, R_1)$ and $F_2 = (A_2, R_2)$ are called *disjoint* if $A_1 \cap A_2 = \emptyset$. Moreover, the union between (not necessarily disjoint) AFs is defined as $F_1 \cup F_2 = (A_1 \cup A_2, R_1 \cup R_2)$.

Example 2.5 *We consider the framework $F = (A, R)$ with $A = \{a, b, c, d, e, f, g, h, i\}$ and $R = \{(a, b), (b, c), (c, a), (b, d), (b, e), (d, f), (e, f), (f, e), (f, g), (g, h), (h, i), (i, f)\}$ as illustrated in Figure 2. F has three SCCs, namely $C_1 = \{a, b, c\}$, $C_2 = \{d\}$ and $C_3 = \{e, f, g, h, i\}$. For example, the argument g belongs to C_3, thus $C_F(g) = C_3$.* ◇

It turns out to be convenient to use two different concepts to obtain sub-frameworks of AFs. Let $F = (A, R)$ be an AF and S a set of arguments. Then, $F|_S = ((A \cap S), R \cap (S \times S))$ is the *sub-framework* of F wrt. S and we also use $F - S = F|_{A \setminus S}$. We note the following relation (which we use implicitly later on), for an AF F and sets S, S': $F|_{S \setminus S'} = F|_S - S' = (F - S')|_S$. In particular, for an AF F, a component $C \in SCCs(F)$ and a set S we thus have $F|_{C \setminus S} = F|_C - S$.

For the framework F of Example 2.5 and the set $S = \{f\}$. Then, $F|_{C_3} - S = (\{e, g, h, i\}, \{(g, h), (h, i)\})$.

We now give the definition of the *cf2* semantics which slightly differs from (but is equivalent to) the original definition in [3, 7]. (i) We use some of the notation established above, like the concept of sub-frameworks and the corresponding relations; (ii) $D_F(S)$, as introduced next, replaces the set "$D_F(S, E)$" and $F|_C - D_F(S)$ replaces "$F_{\downarrow UP_F(S,E)}$"; moreover, the set of undefeated arguments "$U_F(S, E)$" as used in the general schema from [7], is not required here, because the base function for the *cf2* semantics does not make use of this set. Next, we define the set of *component-defeated* arguments $D_F(S)$ which identifies all arguments which are attacked from a given set S from outside their SCC.

Definition 2.6 *Let $F = (A, R)$ be an AF and $S \subseteq A$. An argument $b \in A$ is **component-defeated** by S (in F), if there exists an $a \in S$, such that $(a, b) \in R$ and $a \notin C_F(b)$. The set of arguments component-defeated by S in F is denoted by $D_F(S)$.*

Definition 2.7 *Let $F = (A, R)$ be an argumentation framework and S a set of arguments. Then, S is a cf2 extension of F, i.e. $S \in cf2(F)$, iff*

- *in case $|SCCs(F)| = 1$, then $S \in naive(F)$,*

5

- *otherwise, $\forall C \in SCCs(F)$, $(S \cap C) \in cf2(F|_C - D_F(S))$.*

In words, the recursive definition $cf2(F)$ is based on a decomposition of the AF F into its $SCCs$ depending on a given set S of arguments. We illustrate the behavior of this procedure in the following example.

Example 2.8 *Consider the framework F from Example 2.5. We want to check whether $S = \{a, d, e, g, i\}$ is a $cf2$ extension of F (the arguments of the set S are highlighted in Figure 3). Following Definition 2.7, we first identify the SCCs of F, hence $SCCs(F) = \{C_1, C_2, C_3\}$ as in Example 2.5. Due to the attack (d, f) and $d \in S$ we obtain f as the only component-defeated argument, thus $D_F(S) = \{f\}$. This leads us to the following checks (see also Figure 4 which shows the involved sub-frameworks). Note here that in case $F|_{C_i} - D_F(S) = F|_{C_i}$ we only write $(S \cap C_i) \in cf2(F|_{C_i})$.*

1. *$(S \cap C_1) \in cf2(F|_{C_1})$: $F|_{C_1}$ consists of a single SCC; hence, we have to check whether $(S \cap C_1) = \{a\} \in naive(F|_{C_1})$, which indeed holds.*

2. *$(S \cap C_2) \in cf2(F|_{C_2})$: $F|_{C_2}$ consists of a single argument d (and thus of a single SCC); $(S \cap C_2) = \{d\} \in naive(F|_{C_2})$ thus holds.*

3. *$(S \cap C_3) \in cf2(F|_{C_3} - \{f\})$: $F|_{C_3} - \{f\} = F|_{\{e,g,h,i\}}$ consists of four SCCs, namely $C_4 = \{e\}$, $C_5 = \{g\}$, $C_6 = \{h\}$ and $C_7 = \{i\}$. Hence, we need a second level of recursion for $F' = F|_{\{e,g,h,i\}}$ and $S' = S \cap C_3$. Note that we have $D_{F'}(S') = \{h\}$. The single-argument AFs $F'|_{C_4} = F|_{\{e\}}$, $F'|_{C_5} = F|_{\{g\}}$, $F'|_{C_7} = F|_{\{i\}}$ all satisfy $(S' \cap C_i) \in naive(F'|_{C_i})$; while $F'|_{C_6} - \{h\}$ yields the empty AF. Therefore, $(S' \cap C_6) = \emptyset \in cf2(F|_{C_6} - \{h\})$ holds as well.*

We thus conclude that S is a $cf2$ extension of F. Further $cf2$ extensions of F are $\{b, f, h\}$, $\{b, g, i\}$ and $\{c, d, e, g, i\}$. The extensions of the other semantics for this example are as follows:

- *$stable(F) = \emptyset$;*

- *$adm(F) = \{\{\}, \{g, i\}\}$;*

- *$pref(F) = \{\{g, i\}\}$.*

For the stage semantics we obtain the same result as for the $cf2$ semantics, but this is not the case in general, as we are going to discuss in the next subsection. ◇

2.3 Properties of the $cf2$ Semantics

The $cf2$ semantics has some special properties which clearly differ from the admissible based semantics. Especially the treatment of odd- and even-length cycles is more uniform in the case of $cf2$ semantics.

For our framework of Example 2.5 we obtain $\{g, i\}$ as the only preferred extension. This comes due to the fact that in an odd-length cycle, as is the case in this example for the arguments a, b and c, none of these arguments can be defended. Lets modify the framework in the sense
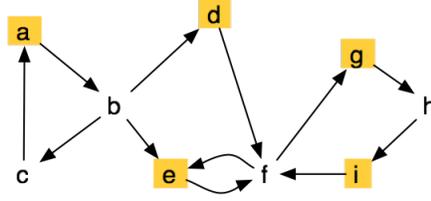
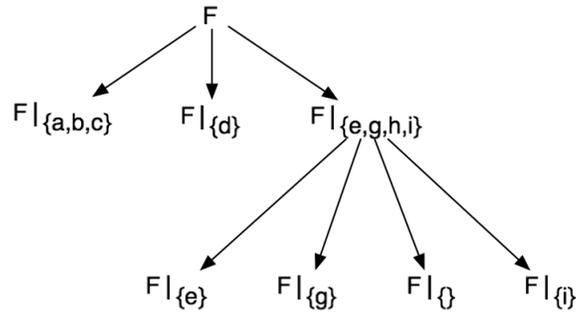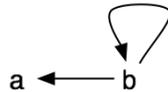Figure 3: The argumentation framework F from Example 2.5.

Figure 4: Tree of recursive calls for computing $cf2(F)$. from Example 2.5.

that we include a new argument x which makes the cycle even, as illustrated in Figure 5. Then, we obtain totally different preferred extensions, namely $\{b, x, g, i\}$, $\{b, x, f, h\}$ and $\{a, c, d, e, g, i\}$ which are conform with the $cf2$ extensions of the modified AF F'. One possible application for the $cf2$ semantics, which makes use of that special behavior, would be for example that we have three agents, let's call them A, B and C, where agent A disagrees with agent B, B disagrees with C and agent C disagrees with agent A. Additionally we have further arguments and attacks as in Figure 2 which are independent from the disagreement of the agents. We would now want to have at least one of the agents to be chosen, which is not possible with the admissible based semantics like preferred. This is exactly what the $cf2$ semantics does by selecting the maximal conflict free sets of the SCC $\{a, b, c\}$. If now there comes a fourth agent into play, let's call him X like in Figure 5, the situation of the whole framework does not change that drastically, we just have four in turn of three agents. But now, we obtain for both semantics, the preferred and the $cf2$ semantics, the same results.

One special case of an odd-length cycle are self-attacking arguments.

Example 2.9 *Consider the following AF F:*

Then, the empty set is the only preferred extension, whereas $\{a\}$ is a cf2 extension. The motivation
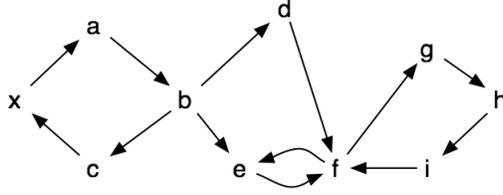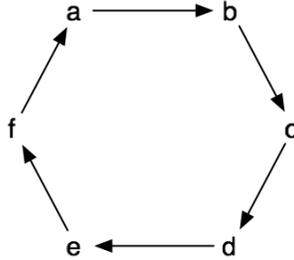
Figure 5: The modified AF F'.

behind selecting $\{a\}$ as a reasonable extension is that it is not necessary to defend a against the attack from b, as b is a self-attacking argument. ◇

Till now, we only mentioned positive properties of the $cf2$ semantics compared to the admissible based semantics. The next example will show a more questionable behavior.

Example 2.10 *Consider the AF F:*

We obtain $stage(F) = pref(F) = stable(F) = \{\{a, c, e\}, \{b, d, f\}\}$, but $cf2(F) = naive(F) = \{\{a, d\}, \{b, e\}, \{c, f\}, \{a, c, e\}, \{b, d, f\}\}$. In this example we have an even-length cycle and the $cf2$ semantics produce three more extensions. This does not really coincide with the motivation for a symmetric treatment of odd- and even-length cycles, as now the results differ significantly for an even-length cycle. ◇

One suggestion to repair the undesired behavior from Example 2.10, could be to check in Definition 2.7 for the case $|SCCs(F)| = 1$ whether $S \in stage(F)$ instead of $S \in naive(F)$. We leave a formalization of this modification for future work.

The relation between the introduced semantics is illustrated in Figure 6, an arrow from semantics σ to semantics τ encodes that each σ-extension is also a τ-extension. The relations between the $cf2$ semantics and the stable, resp. the naive semantics, are due to [6].

As pointed out in Example 2.8, there is no particular relation between the $cf2$ and the preferred semantics, but the stage and the $cf2$ semantics coincide for this framework. The following examples will show that there is no particular relation between stage and $cf2$ extensions as well.

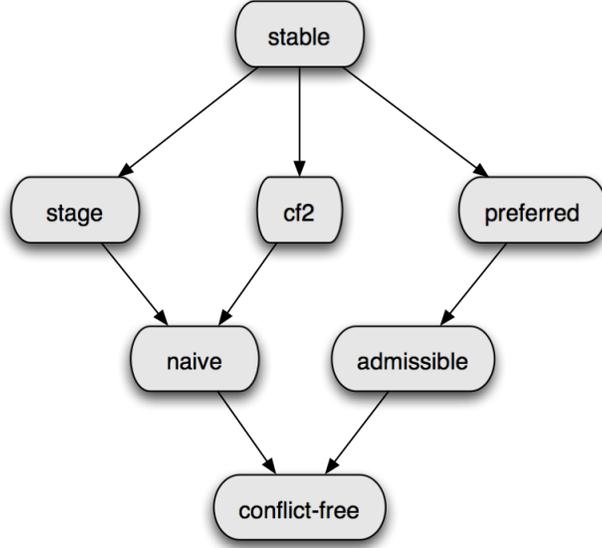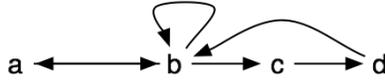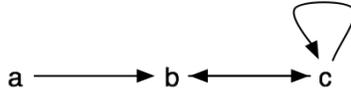**Example 2.11** *Consider the following AF F:*

Figure 6: Relations between Semantics

Here $\{a, c\}$ is the only stage extension of F (it is also stable). Concerning the $cf2$ semantics, note that F is built from a single SCC. Thus, the $cf2$ extensions are given by the maximal conflict-free sets of F, which are $\{a, c\}$ and $\{a, d\}$. Thus, we have $stage(F) \subset cf2(F)$.

As an example for a framework G such that $cf2(G) \subset stage(G)$, consider the following AF:

Then G consists of two SCCs namely $C_1 = \{a\}$ and $C_2 = \{b, c\}$. The conflict-free sets of G are $E_1 = \{a\}$ and $E_2 = \{b\}$. Now it remains to check if E_1 and E_2 are also $cf2$ extensions of G. First we make the check for E_1. Due to the attack (a, b) and $a \in E_1$ we obtain $D_G(E_1) = \{b\}$. We have the following two cases:

- *$(E_1 \cap C_1) \in cf2(G|_{C_1})$, which holds, since $\{a\} \in naive(G|_{C_1})$.*

- *$(E_1 \cap C_2) \in cf2(G|_{C_2} - D_G(E_1))$, which holds, since $\emptyset \in naive(G|_{\{c\}})$.*

For E_2 we obtain $D_G(E_2) = \emptyset$. The check $(E_2 \cap C_1) \in cf2(G|_{C_1})$ does not hold, since $naive(G|_{C_1}) = \{a\}$. Hence, E_2 is not a $cf2$ extension of G. Thus, $cf2(G) = \{E_1\}$ but $stage(G) = \{E_1, E_2\}$. ◇
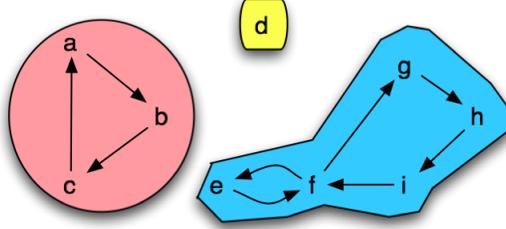
9

Figure 7: Separation of the AF F from Example 2.5.

3 An Alternative Characterization for the $cf2$ Semantics

In the original definition of the SCC-recursive semantics in [7], the computation is based on checking recursively whether a set of arguments fulfills a base function (depending on the semantics) in a single SCC. Thus, the computation is based on a decomposition of the framework along its SCCs. Our alternative characterization is based on the idea to decompose the framework as well, but differently to the original approach the decomposition is only recursive in terms of a certain set of arguments, for which we provide a fixed-point operator. This modification allows us to avoid the recursive computation of several sub-frameworks. Instead we only compute one, possibly not connected, framework where we eliminate the arguments and corresponding attacks which are, what we call, "recursively component defeated".

We start with the following concept.

Definition 3.1 *An AF $F = (A, R)$ is called* **separated** *if for each $(a, b) \in R$, $C_F(a) = C_F(b)$. We define $[[F]] = \bigcup_{C \in SCCs(F)} F|_C$ and call $[[F]]$ the* **separation** *of F.*

In words, an AF is separated if there are no attacks between different strongly connected components. Thus, the separation of an AF always yields a separated AF.

The separation of the framework F of Example 2.5 is depicted in Figure 7. The following technical lemma will be useful later.

Lemma 3.2 *For any AF F and set S of arguments, $\bigcup_{C \in SCCs(F)} [[F|_C - S]] = [[F - S]]$.*

Proof. We first note that for disjoint AFs F and G, $[[F]] \cup [[G]] = [[F \cup G]]$ holds. Moreover, for a set S of arguments and arbitrary frameworks F and G, $(F - S) \cup (G - S) = (F \cup G) - S$ is clear. Using these observations, we obtain

$$\bigcup_{C \in SCCs(F)} [[F|_C - S]] = [[\bigcup_{C \in SCCs(F)} (F|_C - S)]] =$$

$$[[(\bigcup_{C \in SCCs(F)} F|_C) - S]] = [[[[F]] - S]].$$

It remains to show that $[[[[F]] - S]] = [[F - S]]$. Obviously, both AFs possess the same arguments A. Thus, let R be the attacks of $[[[[F]] - S]]$ and R' the attacks of $[[F - S]]$. $R \subseteq R'$ holds by

10

the fact that each attack in $[[F]]$ is also contained in F. To show $R' \subseteq R$, let $(a, b) \in R'$. Then $a, b \notin S$, and $C_{F-S}(a) = C_{F-S}(b)$. From the latter, $C_F(a) = C_F(b)$ and thus (a, b) is an attack in $[[F]]$ and also in $[[F]] - S$. Again using $C_{F-S}(a) = C_{F-S}(b)$, shows $(a, b) \in R$. □

Next, we define the level of recursiveness a framework shows with respect to a set S of arguments and then the aforementioned set of recursively component defeated arguments (by S) in an AF.

Definition 3.3 *For an AF $F = (A, R)$ and a set S of arguments, we recursively define the* **level** $\ell_F(S)$ *of F wrt S as follows:*

- *if $|SCCs(F)| = 1$ then $\ell_F(S) = 1$;*

- *otherwise, $\ell_F(S) = 1 + max(\{\ell_{F|_C - D_F(S)}(S \cap C) \mid C \in SCCs(F)\})$.*

For our running example we obtain the level $\ell_F(S)$ wrt the set $S = \{a, d, e, g, i\}$ as follows. $\ell_F(S) = 1 + max(\{\ell_{F|_C - D_F(S)}(S \cap C) \mid C \in SCCs(F)\})$, where $D_F(S) = \{f\}$ and $SCCs(F) = \{C_1, C_2, C_3\}$ with $C_1 = \{a, b, c\}$, $C_2 = \{d\}$ and $C_3 = \{e, f, g, h, i\}$. This leads to the following recursive calls:

- $\ell_{F|_{C_1}}(S \cap C_1) = 1$,

- $\ell_{F|_{C_2}}(S \cap C_2) = 1$,

- $\ell_{F'}(S') = 1 + max(\{\ell_{F'|_{C'} - D_{F'}(S')}(S' \cap C') \mid C' \in SCCs(F')\})$. Where $F' = F|_{C_3} - D_F(S)$, $S' = S \cap C_3 = \{e, g, i\}$ and $D_{F'}(S') = \{h\}$, furthermore $SCCs(F') = \{C_4, C_5, C_6, C_7\}$ with $C_4 = \{e\}$, $C_5 = \{g\}$, $C_6 = \{h\}$ and $C_7 = \{i\}$. As all those SCCs of F' are single SCCs, we obtain in each recursive call level 1.

To sum up the level of F wrt S is $\ell_F(S) = 3$. One can compare the tree of recursive calls in Figure 3 with the computation of $\ell_F(S)$. When the *height h* of a tree is the length of the path from the root to the deepest node in the tree, we denote the height of the computation tree for the *cf2* semantics for an AF F wrt S as $h_F(S)$, then $\ell_F(S) = h_F(S) + 1$.

Definition 3.4 *Let $F = (A, R)$ be an AF and S a set of arguments. We define the set of arguments* **recursively component defeated** *by S (in F) as follows:*

- *if $|SCCs(F)| = 1$ then $\mathcal{RD}_F(S) = \emptyset$;*

- *otherwise, $\mathcal{RD}_F(S) = D_F(S) \cup \bigcup_{C \in SCCs(F)} \mathcal{RD}_{F|_C - D_F(S)}(S \cap C)$.*

We are now prepared to give our first alternative characterization, which establishes a *cf2* extension S of a given AF F by checking whether S is maximal conflict-free in a certain separated framework constructed from F using S.

Lemma 3.5 *Let $F = (A, R)$ be an AF and S be a set of arguments. Then,*

$$S \in cf2(F) \text{ iff } S \in naive([[F - \mathcal{RD}_F(S)]]).$$

11

Proof. We show the claim by induction over $\ell_F(S)$.

Induction base. For $\ell_F(S) = 1$, we have $|SCCs(F)| = 1$. By definition $\mathcal{RD}_F(S) = \emptyset$ and we have $[[F - \mathcal{RD}_F(S)]] = [[F]] = F$. Thus, the assertion states that $S \in cf2(F)$ iff $S \in naive(F)$ which matches the original definition for the $cf2$ semantics in case the AF has a single strongly connected component.

Induction step. Let $\ell_F(S) = n$ and assume the assertion holds for all AFs F' and sets S' with $\ell_{F'}(S') < n$. In particular, we have by definition that, for each $C \in SCCs(F)$, $\ell_{F|_C - D_F(S)}(S \cap C) < n$. By the induction hypothesis, we thus obtain that, for each $C \in SCCs(F)$, the following holds:

$$(S \cap C) \in cf2(F|_C - D_F(S)) \text{ iff}$$
$$(S \cap C) \in naive\Big([[(F|_C - D_F(S)) - \mathcal{R}'_{F,C,S}]]\Big) \tag{1}$$

where $\mathcal{R}'_{F,C,S} = \mathcal{RD}_{F|_C - D_F(S)}(S \cap C)$. Let us fix now a $C \in SCCs(F)$. Since for each further $C' \in SCCs(F)$ (i.e. $C \neq C'$), no argument from $\mathcal{RD}_{F|_{C'} - D_F(S)}(S \cap C')$ occurs in $F|_C$, we have

$$(F|_C - D_F(S)) - \mathcal{R}'_{F,C,S} =$$
$$\Big((F|_C - D_F(S)) - \mathcal{R}'_{F,C,S}\Big) - \bigcup_{C' \in SCCs(F); C \neq C'} \mathcal{RD}_{F|_{C'} - D_F(S)}(S \cap C') =$$
$$\Big(F|_C - D_F(S)\Big) - \bigcup_{C \in SCCs(F)} \mathcal{RD}_{F|_C - D_F(S)}(S \cap C) =$$
$$F|_C - \Big(D_F(S) \cup \bigcup_{C \in SCCs(F)} \mathcal{RD}_{F|_C - D_F(S)}(S \cap C)\Big) = F|_C - \mathcal{RD}_F(S).$$

Thus, for any $C \in SCCs(F)$, relation (1) amounts to

$$(S \cap C) \in cf2(F|_C - D_F(S)) \text{ iff}$$
$$(S \cap C) \in naive\big([[F|_C - \mathcal{RD}_F(S)]]\big). \tag{2}$$

We now prove the assertion. Let $S \in cf2(F)$. By definition, for each $C \in SCCs(F)$, $(S \cap C) \in cf2(F|_C - D_F(S))$. Using (2), we get that for each $C \in SCCs(F)$, $(S \cap C) \in naive([[F|_C - \mathcal{RD}_F(S)]])$. By the definition of components and the semantics of being maximal conflict-free, the following relation thus follows:

$$\bigcup_{C \in SCCs(F)} (S \cap C) \in naive\Big(\bigcup_{C \in SCCs(F)} [[F|_C - \mathcal{RD}_F(S)]]\Big).$$

Since $S = \bigcup_{C \in SCCs(F)}(S \cap C)$ and, by Lemma 3.2, $\bigcup_{C \in SCCs(F)}[[F|_C - \mathcal{RD}_F(S)]] = [[F - \mathcal{RD}_F(S)]]$, we arrive at $S \in naive([[F - \mathcal{RD}_F(S)]])$ as desired. The other direction is by essentially the same arguments. $\square$

Next, we provide an alternative characterization for $\mathcal{RD}_F(S)$ via a fixed-point operator. In other words, this yields a linearization in the recursive computation of this set. To this end, we require a parametrized notion of reachability.

Definition 3.6 *Let $F = (A, R)$ be an AF, arguments $a, b \in A$ and $B \subseteq A$. We say that b is* **reachable** *in F from a* **modulo** *B, in symbols $a \Rightarrow_F^B b$, if there exists a path from a to b in $F|_B$, i.e. there exists a sequence $c_1, \ldots, c_n$ $(n > 1)$ of arguments such that $c_1 = a$, $c_n = b$, and $(c_i, c_{i+1}) \in R \cap (B \times B)$, for all i with $1 \leq i < n$.*

Definition 3.7 *For an AF $F = (A, R)$, $D \subseteq A$, and a set S of arguments,*

$$\Delta_{F,S}(D) = \{a \in A \mid \exists b \in S : b \neq a, (b, a) \in R, a \not\Rightarrow_F^{A \setminus D} b\}.$$

The operator is clearly monotonic, i.e. $\Delta_{F,S}(D) \subseteq \Delta_{F,S}(D')$ holds for $D \subseteq D'$. As usual, we let $\Delta_{F,S}^0 = \Delta_{F,S}(\emptyset)$ and, for $i > 0$, $\Delta_{F,S}^i = \Delta(\Delta_{F,S}^{i-1})$. Due to monotonicity the least fixed-point (lfp) of the operator exists and, with slightly abuse of notation, will be denoted as $\Delta_{F,S}$. The $\Delta_{F,S}$ operator applied to the empty-set computes recursively the arguments which are defeated from outside their component. Hence, it also takes into account that the SCCs of the framework may change during the computation. We need two more lemmata before showing that $\Delta_{F,S}$ captures $\mathcal{RD}_F(S)$.

Lemma 3.8 *For any AF $F = (A, R)$ and any set $S \subseteq A$, $\Delta_{F,S}^0 = D_F(S)$.*

Proof. We have $\Delta_{F,S}^0 = \Delta_{F,S}(\emptyset) = \{a \in A \mid \exists b \in S : b \neq a, (b, a) \in R, a \not\Rightarrow_F^A b\}$. Hence, $a \in \Delta_{F,S}^0$, if there exists a $b \in S$, such that $(b, a) \in R$ and a does not reach b in F, i.e. $b \notin C_F(a)$. This meets exactly the definition of $D_F(S)$. $\qquad\square$

Lemma 3.9 *For any AF $F = (A, R)$ and any set $S \in cf(F)$,*

$$\Delta_{F,S} = D_F(S) \cup \bigcup_{C \in SCCs(F)} \Delta_{F|_C - D_F(S), (S \cap C)}.$$

Proof. Let $F = (A, R)$. For the $\subseteq$-direction, we show by induction over $i \geq 0$ that $\Delta_{F,S}^i \subseteq D_F(S) \cup \bigcup_{C \in SCCs(F)} \Delta_{F|_C - D_F(S), (S \cap C)}$. To ease notation, we write $\bar{\Delta}_{F,S,C}$ as a shorthand for $\Delta_{F|_C - D_F(S), (S \cap C)}$, where $C \in SCCs(F)$.

Induction base. For $i = 0$, $\Delta_{F,S}^0 \subseteq D_F(S) \cup \bigcup_{C \in SCCs(F)} \bar{\Delta}_{F,S,C}$ follows from Lemma 3.8.

Induction step. Let $i > 0$ and assume $\Delta_{F,S}^j \subseteq D_F(S) \cup \bigcup_{C \in SCCs(F)} \bar{\Delta}_{F,S,C}$ holds for all $j < i$. Let $a \in \Delta_{F,S}^i$. Then, there exists a $b \in S$, such that $(b, a) \in R$ and $a \not\Rightarrow_F^D b$, where $D = A \setminus \Delta_{F,S}^{i-1}$. If $b \notin C_F(a)$, we have also $a \not\Rightarrow_F^A b$ and thus $a \in D_F(S)$. Hence, suppose $b \in C_F(a)$. Then, $a \notin D_F(S)$ and, since $S \in cf(F)$ and $b \in S$, also $b \notin D_F(S)$. Thus, both a and b are contained in the framework $F|_C - D_F(S)$ (and so is the attack (b, a)) for $C = C_F(a)$. Moreover, $b \in (S \cap C)$. Towards a contradiction, assume now $a \notin \bar{\Delta}_{F,S,C}$. This yields that $a \Rightarrow_{F|_C - D_F(S)}^{D'} b$ for $D' = A \setminus \bar{\Delta}_{F,S,C}$, i.e. there exist arguments $c_1, \ldots, c_n$ $(n > 1)$ in $F|_C - D_F(S)$ but not contained in $\bar{\Delta}_{F,S,C}$, such that $c_1 = a$, $c_n = b$, and $(c_i, c_{i+1}) \in R$, for all i with $1 \leq i < n$. Obviously all the c_i's are contained in F as well, but since $a \not\Rightarrow_F^D b$ (recall that $D = A \setminus \Delta_{F,S}^{i-1}$), it must hold that at least one of the c_i's, say c, has to be contained in $\Delta_{F,S}^{i-1}$. By the induction hypothesis, we get $c \in \bar{\Delta}_{F,S,C}$, a contradiction.

13

For the $\supseteq$-direction of the claim we proceed as follows. By Lemma 3.8, we know that $D_F(S) = \Delta_{F,S}^0$ and thus $D_F(S) \subseteq \Delta_{F,S}$. It remains to show that $\bigcup_{C \in SCCs(F)} \Delta_{F|_C - D_F(S),(S \cap C)} \subseteq \Delta_{F,S}$. We show by induction over i that $\Delta_{F|_C - D_F(S),(S \cap C)}^i \subseteq \Delta_{F,S}$ holds for each $C \in SCCs(F)$. Thus, let us fix a $C \in SCCs(F)$ and use $\bar{\Delta}_{F,S,C}^i$ as a shorthand for $\Delta_{F|_C - D_F(S),(S \cap C)}^i$.

Induction base. Let $a \in \bar{\Delta}_{F,S,C}^0$. Then, there is a $b \in (S \cap C)$, such that b attacks a in $F' = F|_C - D_F(S)$ and $a \not\Rightarrow_{F'}^{A'} b$, where A' denotes the arguments of F', i.e. $A' = C \setminus D_F(S)$. Since $F|_C$ is built from a SCC C of F, it follows that $a \not\Rightarrow_F^{A \setminus D_F(S)} b$. Since $b \in S$, $(b,a) \in R$, and $D_F(S) = \Delta_{F,S}^0$ (Lemma 3.8), we get $a \in \Delta_{F,S}^1 \subseteq \Delta_{F,S}$.

Induction step. Let $i > 0$ and assume $\bar{\Delta}_{F,S,C}^j \subseteq \Delta_{F,S}$ for all $j < i$. Let $a \in \bar{\Delta}_{F,S,C}^i$. Then, there is a $b \in (S \cap C)$, such that b attacks a in F' and $a \not\Rightarrow_{F'}^{D'} b$, where $D' = A' \setminus \bar{\Delta}_{F,S,C}^{i-1}$. Towards a contradiction, suppose $a \notin \Delta_{F,S}$. Since $b \in S$ and $(b,a) \in R$, it follows that there exist arguments $c_1, \ldots, c_n$ ($n > 1$) in $F \setminus \Delta_{F,S}$, such that $c_1 = a$, $c_n = b$, and $(c_i, c_{i+1}) \in R$, for all i with $1 \le i < n$. All these c_i's are thus contained in the same component as a, and moreover these c_i's cannot be contained in $D_F(S)$, since $D_F(S) \subseteq \Delta_{F,S}$. Thus, they are contained in $F|_C - D_F(S)$, but since $a \not\Rightarrow_{F'}^{D'} b$, there is at least one such c_i, say c, contained in $\bar{\Delta}_{F,S,C}^{i-1}$. By the induction hypothesis, $c \in \Delta_{F,S}$, a contradiction. $\qquad\square$

We now are able to obtain the desired relation.

Lemma 3.10 *For any AF $F = (A, R)$ and any set $S \in cf(F)$, $\Delta_{F,S} = \mathcal{RD}_F(S)$.*

Proof. The proof is by induction over $\ell_F(S)$.

Induction base. For $\ell_F(S) = 1$, $|SCCs(F)| = 1$ by Definition 3.3. From this and Definition 3.4, we obtain $\mathcal{RD}_F(S) = D_F(S) = \emptyset$. By Lemma 3.8, $\Delta_{F,S}^0 = D_F(S) = \emptyset$. By definition, $\Delta_{F,S} = \emptyset$ follows from $\Delta_{F,S}^0 = \emptyset$.

Induction step. Let $\ell_F(S) = n$ and assume the claim holds for all pairs F', $S' \in cf(F')$, such that $\ell_{F'}(S') < n$. In particular, this holds for $F' = F|_C - D_F(S)$ and $S' = (S \cap C)$, with $C \in SCCs(F)$. Note that $(S \cap C)$ is indeed conflict-free in $F|_C - D_F(S)$. By definition we have, $\mathcal{RD}_F(S) = D_F(S) \cup \bigcup_{C \in SCCs(F)} \mathcal{RD}_{F|_C - D_F(S)}(S \cap C)$ and by Lemma 3.9 we know that $\Delta_{F,S} = D_F(S) \cup \bigcup_{C \in SCCs(F)} \Delta_{F|_C - D_F(S),S \cap C}$. Using the induction hypothesis, i.e. $\Delta_{F|_C - D_F(S),S \cap C} = \mathcal{RD}_{F|_C - D_F(S)}(S \cap C)$, the assertion follows. $\qquad\square$

We finally reached our main result in this section, i.e. an alternative characterization for *cf2* semantics, where the need for recursion is delegated to a fixed-point operator.

Theorem 3.11 *For any AF F, $cf2(F) = \{S \mid S \in cf(F) \cap naive([[F - \Delta_{F,S}]])\}$.*

Proof. The result holds by the following observations. By Lemma 3.5, $S \in cf2(F)$ iff $S \in naive([[F - \mathcal{RD}_F(S)]])$. Moreover, from Lemma 3.10, for any $S \in cf(F)$, $\Delta_{F,S} = \mathcal{RD}_F(S)$. Finally, $S \in cf2(F)$ implies $S \in cf(F)$ (see [7], Proposition 47). $\qquad\square$
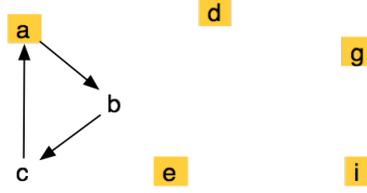
14

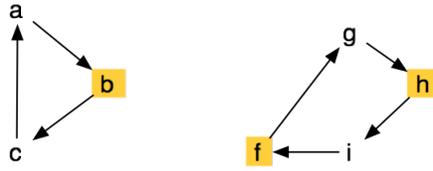Figure 8: Graph of instance $[[F - \Delta_{F,S}]]$ of Example 3.12.

Figure 9: Graph of instance $[[F - \Delta_{F,S'}]]$ of Example 3.12.

Example 3.12 *To exemplify the behavior of $\Delta_{F,S}$ and $[[F - \Delta_{F,S}]]$, we consider the AF F and $S = \{a, d, e, g, i\}$ from Example 2.8. In the first iteration of computing the lfp of $\Delta_{F,S}$, we have $\Delta_{F,S}(\emptyset) = \{f\}$ because the argument f is the only one which is attacked by S but its attacker d is not reachable by f in F. In the second iteration, we obtain $\Delta_{F,S}(\{f\}) = \{f, h\}$, and in the third iteration we reach the lfp with $\Delta_{F,S}(\{f, h\}) = \{f, h\}$. Hence, $[[F - \Delta_{F,S}]]$ of the AF F wrt S is given by*

$$[[F - \Delta_{F,S}]] = \big(\{a, b, c, d, e, g, i\}, \{(a, b), (b, c), (c, a)\}\big).$$

Figure 8 shows the graph of $[[F - \Delta_{F,S}]]$. It is easy to see that $S \in naive([[F - \Delta_{F,S}]])$ as expected, since $S \in cf2(F)$. For comparison, Figure 9 shows the graph of $[[F - \Delta_{F,S'}]]$ wrt the cf2 extension $S' = \{b, f, h\}$ consisting of two SCCs. ◇

4 Complexity Analysis

In this section we investigate the computational complexity of the $cf2$ semantics. We consider the following decision problems for given $F = (A, R)$, $a \in A$ and $S \subseteq A$:

- Ver_{cf2}: is $S \in cf2(F)$?

- Cred_{cf2}: is a contained in at least one $cf2$ extension of F?

- Skept_{cf2}: is a contained in every $cf2$ extension of F?

- NE_{cf2}: is there any $S \in cf2(F)$ for which $S \neq \emptyset$?

So far, the only mentionable reference in this context is the article of Nieves et al. [19], where the authors state that the decision problem Ver_{cf2} is in P. In the following we proof this statement with the help of our alternative characterization.
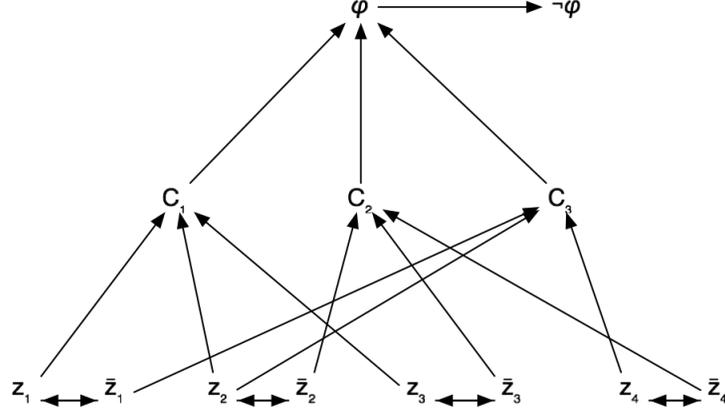
Figure 10: AF F_φ for the example 3-CNF φ.

Theorem 4.1 Ver_{cf2} *is in* P.

Proof. For any AF $F = (A, R)$ and a set $S \subseteq A$, to check if $S \in cf2(F)$ can be computed in polynomial time. We show that all steps in Definition 3.11 are in P. Verifying if $S \in cf(F)$ and $S \in naive(F)$ can be done in polynomial time. Given $\Delta_{F,S}$, computing the instance $[[F - \Delta_{F,S}]]$ can be done efficiently; this follows from known results about graph reachability and efficient algorithms for computing SCCs [22]. It remains to show that the operator $\Delta_{F,S}(D)$ reaches its fixed-point after a polynomial number of iterations. The operator is clearly monotonic, and it is easy to see that in every iteration less or equal connections between the arguments do exist. Hence, the computation terminates when no argument a is attacked from any $b \in S$, and $a \not\Rightarrow_F^{A \setminus D} b$. $\qquad \square$

For the hardness proofs of Cred_{cf2} and Skept_{cf2} we use the standard reduction from propositional formulas in conjunctive normal form (CNF) to AFs as in [10, 13].

Definition 4.2 *Given a 3-CNF formula* $\varphi = \bigwedge_{j=1}^m C_j$ *over atoms* Z *with* $C_j = l_{j1} \vee l_{j2} \vee l_{j3}$ *($1 \le j \le m$) the corresponding AF* $F_\varphi = (A_\varphi, R_\varphi)$ *is built as follows:*

$$
\begin{aligned}
A_\varphi \;&=\; Z \cup \bar{Z} \cup \{C_1, \ldots, C_m\} \cup \{\varphi\} \cup \{\neg\varphi\} \\
R_\varphi \;&=\; \{(z, \bar{z}), (\bar{z}, z) \mid z \in Z\} \cup \{(C_j, \varphi) \mid j \in \{1, \ldots, m\}\} \cup \{(\varphi, \neg\varphi)\} \cup \\
&\qquad \{(z, C_j) \mid j \in \{1, \ldots, m\}, z \in \{l_{j1}, l_{j2}, l_{j3}\}\} \cup \\
&\qquad \{(\bar{z}, C_j) \mid j \in \{1, \ldots, m\}, \neg z \in \{l_{j1}, l_{j2}, l_{j3}\}\}
\end{aligned}
$$

Figure 10 illustrates the AF F_φ *for the formula* $\varphi = (z_1 \vee z_2 \vee z_3) \wedge (\neg z_2 \vee \neg z_3 \vee \neg z_4) \wedge (\neg z_1 \vee z_2 \vee z_4)$.

Lemma 4.3 *For any cf2 extension* E *of the AF* $F_\varphi = (A_\varphi, R_\varphi)$ *and* $z_i \in Z$ *for* $i \in \{1, \ldots, n\}$, *either* $z_i \in E$ *or* $\bar{z}_i \in E$.

Proof. The AF F_φ has the following singleton SCCs $\{\varphi\}$, $\{\neg\varphi\}$, and C_i ($1 \le i \le m$). The remaining SCCs are $C_{l_i} \in \{C_{l_1}, \ldots, C_{l_n}\}$, with $C_{l_i} = \{z_i, \bar{z}_i\}$. As all C_{l_i} are not attacked from outside their component they remain unchanged in $[[F_\varphi - \Delta_{F_\varphi, E}]]$ and $naive(F_\varphi|_{C_{l_i}}) = \{\{z_i\}, \{\bar{z}_i\}\}$. Hence, either $z_i \in E$ or $\bar{z}_i \in E$ (but never both). □

Theorem 4.4 Cred_{cf2} *is NP-complete.*

Proof. For hardness, we show that any 3-CNF formula φ is satisfied iff the corresponding AF F_φ as in Definition 4.2 has a *cf2* extension containing φ.

For the if direction, let φ be a 3-CNF formula over Z and $M \subseteq Z$ a model of φ. We show that $E = \{\{z_i \mid z_i \in M\} \cup \{\bar{z}_i \mid z_i \in Z \setminus M\} \cup \{\varphi\}\}$ is a *cf2* extension of F_φ. We need to show that (i) E is conflict free in F_φ and (ii) $E \in naive([[F_\varphi - \Delta_{F_\varphi, E}]])$. As to (i), from Lemma 4.3 we know that for all $i \in \{1, \ldots, n\}$ either z_i or $\bar{z}_i$ is in E, so there are no conflicts between the arguments in Z and $\bar{Z}$. The argument φ is not attacked by any z_i at all. Hence, $E \in cf(F_\varphi)$. As to (ii), let us first compute $\Delta_{F_\varphi, E}$, where $\Delta_{F_\varphi, E}(\emptyset) = \{x \in A_\varphi \mid \exists l \in E : l \ne x, (l, x) \in R_\varphi, x \not\Rightarrow l\}$. As M is a model of φ, all clauses in φ are satisfied, hence, $\forall C_j \exists l_i$ such that $(l_i, C_j) \in R_\varphi$, where $l_i \in \{z_i, \bar{z}_i\}$ for $j = \{1, \ldots, m\}$ and $i = \{1, \ldots, n\}$. Furthermore, $\varphi \in E$, $(\varphi, \neg\varphi) \in R_\varphi$ and $\neg\varphi \not\Rightarrow \varphi$. Therefore, we obtain $\Delta_{F_\varphi, E}(\emptyset) = \{C_1, \ldots, C_m, \neg\varphi\}$ which is also the lfp $\Delta_{F_\varphi, E}$. Finally, we compute the instance $[[F_\varphi - \Delta_{F_\varphi, E}]] = (A_\varphi \setminus \{C_1, \ldots, C_m, \neg\varphi\}, \{(z, \bar{z}), (\bar{z}, z) \mid z \in Z\})$. It is easy to see that $E \in naive([[F_\varphi - \Delta_{F_\varphi, E}]])$ holds.

Only if: Let $E \in cf2(F_\varphi)$ such that $\varphi \in E$. We show that $M = \{z_i \mid z_i \in E\} \cup \{\neg z_i \mid \bar{z}_i \in E\}$ is a model of φ. As $\varphi \in E$ we know that it is not attacked by any $d \in \Delta_{F_\varphi, E}$. Assume there exists a $C_j \notin \Delta_{F_\varphi, E}$ with $(C_j, \varphi) \in R_\varphi$. We know $C_j \notin E$ because $E \in cf(F_\varphi)$, hence from Definition 3.7 we conclude there is no $x \in E$ such that $(x, C_j) \in R_\varphi$. In this case, the argument C_j is contained in $[[F_\varphi - \Delta_{F_\varphi, E}]]$, but this is a contradiction to $E \in naive([[F_\varphi - \Delta_{F_\varphi, E}]])$, because the set $E' = E \cup \{C_j\}$ is conflict-free in $[[F_\varphi - \Delta_{F_\varphi, E}]]$. It follows that for each C_j there exists a $l_i \in \{z_i, \bar{z}_i\}$ such that $(l_i, C_j) \in R_\varphi$, for $j = \{1, \ldots, m\}$. This means that for every clause C_j there exists a literal $l_i \in M$. Hence, M is a model of φ.

For membership one can construct an algorithm as follows. For any AF $F = (A, R)$ and $a \in A$, guess $S \subseteq A$ with $a \in S$ and check $S \in cf2(F)$. As $\mathrm{Ver}_{cf2} \in$ P, this yields an NP algorithm. □

Theorem 4.5 Skept_{cf2} *is coNP-complete.*

Proof. For hardness, we show that a given 3-CNF formula φ is unsatisfiable iff $\neg\varphi$ is contained in every *cf2* extension of F_φ, where F_φ is constructed following Definition 4.2. From the proof of Theorem 4.4, we already know that φ is contained in a *cf2* extension iff φ is satisfiable. By definition of the *cf2* semantics, it is easily seen that each *cf2* extension of F_φ which does not contain argument φ, has to contain $\neg\varphi$. Thus, in case φ is unsatisfiable, argument $\neg\varphi$ is indeed skeptically accepted.

Membership can be shown as follows via the complementary problem. Thus, for given AF $F = (A, R)$ and $a \in A$ we guess a set S with $a \notin S$ and check $S \in cf2(F)$. As $\mathrm{Ver}_{cf2} \in$ P, this yields an NP algorithm for the complementary problem of Skept_{cf2}. Hence, Skept_{cf2} is in coNP. □

	cf2	stable	stage	adm	pref
Ver_σ	in P	in P	coNP-c	in P	coNP-c
Cred_σ	NP-c	NP-c	Σ_2^P-c	NP-c	NP-c
Skept_σ	coNP-c	coNP-c	Π_2^P-c	Trivial	Π_2^P-c
NE_σ	in P	NP-c	in L	NP-c	NP-c

Table 1: Complexity of decision problems ($\mathcal{C}$-c denotes completeness for class $\mathcal{C}$).

Theorem 4.6 $\mathrm{NE}_{cf2} \in P$

Proof. Recall, that for every AF F it holds that each *cf2* extension of F is also a naive extension of F. Thus, in case we have that F possesses only the empty set as its *cf2* extension, we know that the empty set is also the only naive extension of F. However, this is only the case if all arguments of F are self-attacking. Thus to decide whether there exists a nonempty *cf2* extension, of an AF $F = (A, R)$, it is sufficient to check if there exists any argument $a \in A$ such that $(a, a) \notin R$. This can be done in polynomial time. □

Our results are summarized in the first column of Table 1 together with results of the other semantics used in this context ([10, 13, 14, 15]). We observe that the complexity of the *cf2* semantics behaves slightly different to these semantics.

5 Strong Equivalence of Argumentation Semantics

So far, we have focused exclusively on the *cf2* semantics. In this section, we will show a distinguished feature of the *cf2* semantics, which separates it from all other important semantics proposed for abstract argumentation. In a nutshell, this particular property states that each attack in an AF has a potential "meaning" under the *cf2* semantics, while this is not the case for other semantics where attacks may be redundant as the following example illustrates.

Example 5.1 *Consider the following AFs F and G.*

For both AFs there does not exist any stable extension, hence $stable(F) = stable(G) = \emptyset$. Now, we add the new AF $H = (\{b, e\}, \{(b, e)\})$, then they still have the same stable extensions $stable(F \cup H) = stable(G \cup H) = \{\{b, d\}\}$, as highlighted in the following graphs.

In fact, the attacks $\{(a,b), (e,b)\}$ in F as well as the attacks $\{(a,d), (e,c), (e,d)\}$ in G are redundant under the stable semantics. Furthermore, it can be shown that no matter which framework H one adds to F and G they will always posses the same stable extensions. ◇

The concept described in Example 5.1 has been analyzed by Oikarinen and Woltran in [20] and is defined as *strong equivalence*.

Definition 5.2 *Two AFs F and G are* **strongly equivalent** *to each other wrt. a semantics σ, in symbols $F \equiv_s^\sigma G$, iff for each AF H, $\sigma(F \cup H) = \sigma(G \cup H)$.*

By definition, $F \equiv_s^\sigma G$ implies $\sigma(F) = \sigma(G)$, but the other direction is not true in general.

In what follows, we show that for the *cf2* semantics strong equivalence coincides with syntactic equivalence. Afterwards, we characterize strong equivalence for the stage and naive semantics, both of them have not been considered in [20].

5.1   Strong Equivalence wrt. *cf2* Semantics

Interestingly, it turns out that for this semantics there are no redundant attacks at all. In fact, even in the case where an attack links two self-attacking arguments, this attack might play a role by gluing two components together. Having no redundant attacks means that strong equivalence coincides with syntactic equivalence. Before we give the next theorem we provide two lemmata which will also be useful later.

Lemma 5.3 *For any AFs F and G with $A(F) \neq A(G)$, there exists an AF H such that $A(H) \subseteq A(F) \cup A(G)$ and $\sigma(F \cup H) \neq \sigma(G \cup H)$, for the semantics $\sigma \in \{naive, stage, cf2\}$.*

Proof. In case $\sigma(F) \neq \sigma(G)$, we just consider $H = (\emptyset, \emptyset)$ and get $\sigma(F \cup H) \neq \sigma(G \cup H)$. Thus assume $\sigma(F) = \sigma(G)$ and let wlog. $a \in A(F) \setminus A(G)$. Thus for all $E \in \sigma(F)$, $a \notin E$. Consider the framework $H = (\{a\}, \emptyset)$. Then, for all $E' \in \sigma(G \cup H)$, we have $a \in E'$. On the other hand, $F \cup H = F$ and also $\sigma(F \cup H) = \sigma(F)$. Hence, a is not contained in any $E \in \sigma(F \cup H)$, and we obtain $\sigma(F \cup H) \neq \sigma(G \cup H)$. □

Lemma 5.4 *For any AFs F and G such that $(a,a) \in R(F) \setminus R(G)$ or $(a,a) \in R(G) \setminus R(F)$, there exists an AF H such that $A(H) \subseteq A(F) \cup A(G)$ and $\sigma(F \cup H) \neq \sigma(G \cup H)$, for $\sigma \in \{naive, stage, cf2\}$.*

Figure 11: $F \cup H$
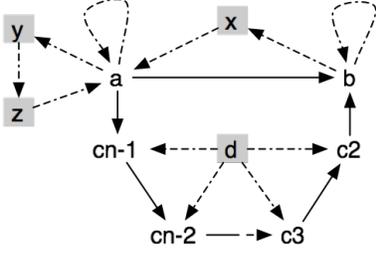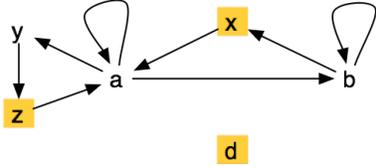

Figure 12: $G \cup H$

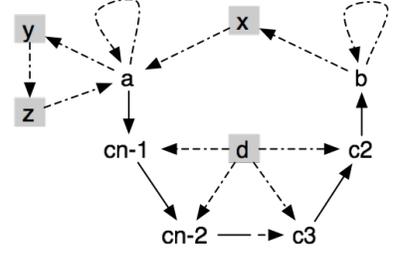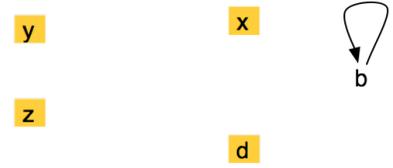Figure 13: $[[(F \cup H) - \Delta_{F \cup H, E}]]$

Figure 14: $[[(G \cup H) - \Delta_{G \cup H, E}]]$

Proof. Let the self-attack $(a, a) \in R(F) \setminus R(G)$ and consider the framework $H = (A, \{(a, b), (b, b) \mid a, b \in A, a \neq b\})$ with $A = A(F) \cup A(G)$. Then $\sigma(G \cup H) = \{a\}$ while $\sigma(F \cup H) = \{\emptyset\}$ for all considered semantics $\sigma \in \{naive, stage, cf2\}$. For example, in case $\sigma = cf2$ we obtain $\Delta_{G \cup H, E} = \{b \mid b \in A \setminus \{a\}\}$. Moreover, $\{a\}$ is conflict-free in $G \cup H$ and $\{a\} \in naive(G')$, where $G' = (G \cup H) - \Delta_{G \cup H, E} = (\{a\}, \emptyset)$. On the other hand, $cf2(F \cup H) = \{\emptyset\}$ since all arguments in $F \cup H$ are self-attacking. The case for $(a, a) \in R(G) \setminus R(F)$ is similar. $\qquad \square$

Theorem 5.5 *For any AFs F and G, $F \equiv_s^{cf2} G$ iff $F = G$.*

Proof. Since for any AFs $F = G$ obviously implies for all AFs H, $cf2(F \cup H) = cf2(G \cup H)$, we only have to show that if $F \neq G$ there exists an AF H such that $cf2(F \cup H) \neq cf2(G \cup H)$. From Lemma 5.3 and Lemma 5.4 we know that in case the arguments or the self-loops are not equal in both frameworks, there exists an AF H such that $cf2(F \cup H) \neq cf2(G \cup H)$. We thus assume that $A = A(F) = A(G)$ and $(a, a) \in R(F)$ iff $(a, a) \in R(G)$, for each $a \in A$. Let us thus suppose wlog. an attack $(a, b) \in R(F) \setminus R(G)$ and consider the AF

$$H = (A \cup \{d, x, y, z\}, \{(a, a), (b, b), (b, x), (x, a), (a, y), (y, z), (z, a),$$
$$(d, c) \mid c \in A \setminus \{a, b\}\}),$$

see also Figures 11 and 12 for illustration. Then, there exists a set $E = \{d, x, z\}$, such that $E \in cf2(F \cup H)$ but $E \notin cf2(G \cup H)$. To show that $E \in cf2(F \cup H)$, we first compute $\Delta_{F \cup H, E} = \{c \mid c \in A \setminus \{a, b\}\}$. Thus, in the instance $[[(F \cup H) - \Delta_{F \cup H, E}]]$ we have two SCCs left, namely $C_1 = \{d\}$ and $C_2 = \{a, b, x, y, z\}$ as illustrated in Figure 13. Furthermore, all attacks between the arguments of C_2 are preserved, and we obtain that $E \in naive([[(F \cup H) - \Delta_{F \cup H, E}]])$, and as it is also conflict-free we have that $E \in cf2(F \cup H)$ as well. On the other hand, we obtain

20

$\Delta_{G \cup H, E} = \{a\} \cup \{c \mid c \in A \setminus \{a, b\}\}$, and the instance $G' = [[(G \cup H) - \Delta_{G \cup H, E}]]$ consists of five SCCs, namely $C_1 = \{d\}$, $C_2 = \{b\}$, $C_3 = \{x\}$, $C_4 = \{y\}$ and $C_5 = \{z\}$, with b being self-attacking as illustrated in Figure 14. Thus, the set $E' = \{d, x, y, z\} \supset E$ is conflict-free in G'. Therefore, we obtain $E \notin naive(G')$, and hence, $E \notin cf2(G \cup H)$. $F \not\equiv_s^{cf2} G$ follows. $\square$

In other words, the proof of Theorem 5.5 shows that no matter which AFs $F \neq G$ are given, we can always construct a framework H such that $cf2(F \cup H) \neq cf2(G \cup H)$. In particular, we can always add new arguments and attacks such that the missing attack in one of the original frameworks leads to different SCCs in the modified ones and therefore to different $cf2$ extensions, when suitably augmenting the two AFs under comparison.

The $cf2$ semantics is the only semantics considered so far, where strong equivalence coincides with syntactic equivalence. This can be seen as another special property of the $cf2$ semantics which gives raise to a more formal investigation. Therefore, we introduce a new property for argumentation semantics which we call the *succinctness property*. In contrast to strong equivalence which considers particular AFs, the succinctness property denotes a general property for argumentation semantics and is comparable to the evaluation criteria proposed in [4]. Hence, it is independent of the specific instantiation method.

In Example 5.1 we already talked informally about redundant attacks in AFs, in the next definition we make this idea formal; for AFs $F = (A, R)$ and $F' = (A', R')$ we write $F \subseteq F'$ to denote that $A \subseteq A'$ and $R \subseteq R'$ jointly hold. Moreover, we use $F \setminus (a, b)$ as a shorthand for the framework $(A, R \setminus \{(a, b)\})$.

Definition 5.6 *For an AF $F = (A, R)$ and semantics σ we call an attack $(a, b) \in R$ **redundant in** F **wrt.** σ if for all F' with $F \subseteq F'$, $\sigma(F') = \sigma(F' \setminus (a, b))$.*

The succinctness property identifies to which extend attacks contribute in terms of a given semantics. In other words, we are interested here in how many attacks are possibly ignored in the computation of a semantics. The concept of succinctness is now captured as follows.

Definition 5.7 *An argumentation semantics σ satisfies the **succinctness property** or is **maximal succinct** iff no AF contains a redundant attack wrt. σ.*

The following theorem gives the link between the succinctness property and strong equivalence.[1]

Theorem 5.8 *An argumentation semantics σ satisfies the succinctness property iff for any AFs F, G with $A(F) = A(G)$ it holds that strong equivalence between F and G wrt. σ ($F \equiv_s^\sigma G$) coincides with syntactic equivalence, i.e. $F = G$.*

---

[1]The result as stated here requires $A(F) = A(G)$, since we do not impose any prerequisites on the given semantics σ. However, in case $A(F) \neq A(G)$ implies $F \not\equiv_s^\sigma G$ (as it is the case for all semantics studied in terms of strong equivalence so far, see also Lemma 5.3), the result is even stronger saying that succinctness holds, if and only if, strong and syntactic equivalence coincide.

Proof. Suppose σ does not satisfy the the succinctness property, i.e. there exists an AF F and an attack (a, b) in F such that $\sigma(F \cup H) = \sigma((F \setminus (a, b)) \cup H)$ for any AF H. Obviously, F and $F \setminus (a, b)$ are given over the same arguments and we have $F \equiv_s^\sigma F \setminus (a, b)$ but $F \neq F \setminus (a, b)$.

Suppose we have AFs F, G with $A(F) = A(G)$ such that $F \neq G$ and $F \equiv_\sigma^s G$. W.l.o.g. let (a, b) be an attack in F which does not occur in G. Since $F \equiv_\sigma^s G$, $\sigma(F \cup H) = \sigma(G \cup H)$, in particular for all H not containing (a, b). Since $F \cup H \cup (a, b) = F \cup H$, we get that $\sigma(G \cup (a, b) \cup H) = \sigma(G \cup H)$ for all H. By setting $G' = G \cup (a, b)$, we observe that (a, b) is redundant in G' wrt. σ. Hence, σ cannot be maximal succinct. □

From Theorem 5.5 and Theorem 5.8 we conclude that the *cf2* semantics satisfies the succinctness property.

5.2 Strong Equivalence wrt. other Semantics

To complete the picture about strong equivalence and succinctness, we give characterizations for strong equivalence wrt. the stage and the naive semantics. As it turns out, these characterizations will be different to syntactical equivalence. By Theorem 5.8, these two semantics are thus not maximal succinct. Recall that the results in [20] in combination with Theorem 5.8 show that many other semantics are not maximal succinct, as well.

To characterize strong equivalence, Oikarinen and Woltran used in [20] so-called *kernels* for different semantics which implicitly remove the redundant attacks of the compared frameworks. As shown in [20], deciding strong equivalence then amounts to checking the syntactic equivalence of the kernels of the two compared frameworks. More precisely, such kernels have been provided for many important semantics, viz. for admissible, preferred, ideal, semi-stable, eager, complete and grounded semantics. All these kernels are non-trivial in the sense that certain attacks are removed.

In order to characterize strong equivalence wrt. stage semantics, we require here exactly the same kernel as already used in [20] to characterize strong equivalence wrt. stable semantics.

Example 5.9 *Consider the frameworks F and G:*

They only differ in the attacks outgoing from the argument a which is self-attacking and yield the same single stage extension, namely $\{c\}$, for both frameworks. We can now add, for instance, $H = (\{a, c\}, \{(c, a)\})$ and the stage extensions for $F \cup H$ and $G \cup H$ still remain the same. In fact, no matter how H looks like, $stage(F \cup H) = stage(G \cup H)$ will hold. ◇

The following kernel reflects the intuition given in the previous example.

Definition 5.10 *For an AF $F = (A, R)$, define $F^{sk} = (A, R^{sk})$ where*

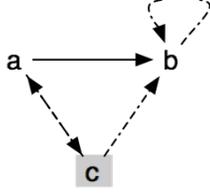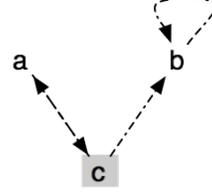$$R^{sk} = R \setminus \{(a, b) \mid a \neq b, (a, a) \in R\}.$$

Figure 15: $F \cup H$

Figure 16: $G \cup H$

Theorem 5.11 *For any AFs F and G, $F \equiv_s^{stage} G$ iff $F^{sk} = G^{sk}$.*

Proof. Only-if: Suppose $F^{sk} \neq G^{sk}$, we show that $F \not\equiv_s^{stage} G$. From Lemma 5.3 and Lemma 5.4 we know that in case the arguments or the self-loops are not equal in both frameworks, $F \equiv_s^{stage} G$ does not hold. We thus assume that $A = A(F) = A(G)$ and $(a,a) \in F$ iff $(a,a) \in G$, for each $a \in A$. Let thus wlog. $(a,b) \in F^{sk} \setminus G^{sk}$. We can conclude $(a,b) \in F$ and $(a,a) \notin F$, thus $(a,a) \notin G$ and $(a,b) \notin G$. Let c be a fresh argument and take

$$H = \{A \cup \{c\}, \{(b,b)\} \cup \{(c,d) \mid d \in A\} \cup \{(a,d) \mid d \in A \cup \{c\} \setminus \{b\}\}).$$

Then, $\{a\}$ is a stage extension of $F \cup H$ (it attacks all other arguments) but not of $G \cup H$ (b is not attacked by $\{a\}$); see also Figures 15 and 16 for illustration.

For the if-direction, suppose $F^{sk} = G^{sk}$. Let us first show that $F^{sk} = G^{sk}$ implies $cf(F \cup H) = cf(G \cup H)$, for each AF H. Towards a contradiction, suppose such an H exists and wlog. let $T \in cf(F \cup H) \setminus cf(G \cup H)$. Since $F^{sk} = G^{sk}$, we know $A(F) = A(G)$. Thus there exist $a, b \in T$ (not necessarily $a \neq b$) such that $(a,b) \in G \cup H$ or $(b,a) \in G \cup H$. On the other hand $(a,b) \notin F \cup H$ and $(b,a) \notin F \cup H$ hold since $a, b \in T$ and $T \in cf(F \cup H)$). Thus, in particular, $(a,b) \notin F$ and $(b,a) \notin F$ as well as $(a,b) \notin H$ and $(b,a) \notin H$; the latter implies $(a,b) \in G$ or $(b,a) \in G$. Suppose $(a,b) \in G$ (the other case is symmetric). If $(a,a) \in G$ then $(a,a) \in G^{sk}$, but $(a,a) \notin F^{sk}$ (since $a \in T$ and thus $(a,a) \notin F$). If $(a,a) \notin G$, $(a,b) \in G^{sk}$ but $(a,b) \notin F^{sk}$ (since $(a,b) \notin F$). In either case $F^{sk} \neq G^{sk}$, a contradiction.

We next show that $F^{sk} = G^{sk}$ implies $(F \cup H)^{sk} = (G \cup H)^{sk}$ for any AF H. Thus, let $(a,b) \in (F \cup H)^{sk}$, and assume $F^{sk} = G^{sk}$; we show $(a,b) \in (G \cup H)^{sk}$. Since, $(a,b) \in (F \cup H)^{sk}$ we know that $(a,a) \notin F \cup H$ and therefore, $(a,a) \notin F^{sk}$, $(a,a) \notin G^{sk}$ and $(a,a) \notin H^{sk}$. Hence, we have either $(a,b) \in F^{sk}$ or $(a,b) \in H^{sk}$. In the later case, $(a,b) \in (G \cup H)^{sk}$ follows because $(a,a) \notin G^{sk}$ and $(a,a) \notin H^{sk}$. In case $(a,b) \in F^{sk}$, we get by the assumption $F^{sk} = G^{sk}$, that $(a,b) \in G^{sk}$ and since $(a,a) \notin H^{sk}$ it follows that $(a,b) \in (G \cup H)^{sk}$.

Finally we show that for any frameworks K and L such that $K^{sk} = L^{sk}$, and any $S \in cf(K) \cap cf(L)$, $S_R^+(K) = S_R^+(L)$. This follows from the fact that for each $s \in S$, (s,s) is neither contained in K nor in L. But then each attack $(s,b) \in K$ is also in K^{sk}, and likewise, each attack $(s,b) \in L$ is also in L^{sk}. Now since $K^{sk} = L^{sk}$, $S_R^+(K) = S_R^+(L)$ is obvious.

Thus, we showed that, given $F^{sk} = G^{sk}$, the following relations hold for each AF H: $cf(F \cup H) = cf(G \cup H)$; $(F \cup H)^{sk} = (G \cup H)^{sk}$; and $S_R^+(F \cup H) = S_R^+(G \cup H)$ holds for each $S \in cf(F \cup H) = cf(G \cup H)$ (taking $K = F \cup H$ and $L = G \cup H$). Thus, $stage(F \cup H) = stage(G \cup H)$, for each AF H. Consequently, $F \equiv_s^{stage} G$. $\qquad\square$

23

The second semantics we consider here is the naive semantics, as it is closely related to the $cf2$ semantics and has not yet been included in [20]. Here, strong equivalence is only a marginally more restricted concept than standard equivalence, namely in case the two compared AFs are not given over the same arguments. To proof the next theorem we need the following lemma.

Lemma 5.12 *Let F and H be AFs and S be a set of arguments. Then, $S \in cf(F \cup H)$ iff, jointly $(S \cap A(F)) \in cf(F)$ and $(S \cap A(H)) \in cf(H)$.*

Proof. The only-if direction is clear. Thus suppose $S \notin cf(F \cup H)$. Then, there exist $a, b \in S$, such that $(a, b) \in F \cup H$. By our definition of "$\cup$", then $(a, b) \in F$ or $(a, b) \in H$. But then $(S \cap A(F)) \notin cf(F)$ or $(S \cap A(H)) \notin cf(H)$ follows. □

Theorem 5.13 *The following statements are equivalent: (1) $F \equiv_s^{naive} G$; (2) $naive(F) = naive(G)$ and $A(F) = A(G)$; (3) $cf(F) = cf(G)$ and $A(F) = A(G)$.*

Proof. (1) implies (2): basically by the definition of strong equivalence and Lemma 5.3.

(2) implies (3): Assume $naive(F) = naive(G)$ but $cf(F) \neq cf(G)$. Wlog. let $S \in cf(F) \setminus cf(G)$. Then, there exists a set $S' \supseteq S$ such that $S' \in naive(F)$ and by assumption then $S' \in naive(G)$. However, as $S \notin cf(G)$ there exists an attack $(a, b) \in R(G)$, such that $a, b \in S$. But as $S \subseteq S'$, we have $S' \notin cf(G)$ as well; a contradiction to $S' \in naive(G)$.

(3) implies (1): Suppose $F \not\equiv_s^{naive} G$, i.e. there exists a framework H such that $naive(F \cup H) \neq naive(G \cup H)$. Wlog. let now $S \in naive(F \cup H) \setminus naive(G \cup H)$. From Lemma 5.12 one can show that $(S \cap A(F)) \in naive(F)$ and $(S \cap A(H)) \in naive(H)$, as well as $(S \cap A(G) \notin naive(G)$. Let us assume $S' = S \cap A(F) = S \cap A(G)$, otherwise we are done yielding $A(F) \neq A(G)$. If $S' \notin cf(G)$ we are also done (since $S' \in cf(F)$ follows from $S' \in naive(F)$); otherwise, there exists an $S'' \supset S'$, such that $S'' \in cf(G)$. But $S'' \notin cf(F)$, since $S' \in naive(F)$. Again we obtain $cf(F) \neq cf(G)$ which concludes the proof. □

It follows that the stage and naive semantics are not maximal succinct due to Theorem 5.8.

6 Discussion

In this paper, we investigated the $cf2$ semantics in several ways. We introduced an alternative characterization which is based on a certain fixed-point operator in order to avoid the more involved recursions from the original definition [7]. With this new characterization we were able to prove complexity results for the reasoning problems Ver_{cf2}, NE_{cf2}, Cred_{cf2} and Skept_{cf2}. While the former two problems can be decided in polynomial time we obtained that Cred_{cf2} is NP-complete and Skept_{cf2} is coNP-complete. Except for NE_{cf2}, the $cf2$ semantics thus has the same complexity as the stable semantics. Let us mention here also that the behavior of the $cf2$ semantics is similar to the stage semantics. Both semantics are based on the computation of conflict-free sets, they can select arguments out of an odd-length cycle and they are not influenced by self-attacking arguments. However, for the stage semantics both, credulous and skeptical acceptance is on the second level of the polynomial hierarchy. This mirrors the benefit of using graph-theoretic properties for the

computation of extensions. Furthermore, it shows that the graph representation is not only useful for illustrating the frameworks, but also for using the structural properties in the characterization of the semantics. Additionally, our new characterization of the *cf2* semantics allowed us to provide a relatively compact encoding for computing *cf2* extensions in terms of logic programs; these reductions have been incorporated to the ASP-based argumentation system ASPARTIX, see [16, 17] for details.

The newly introduced succinctness property allows to relate the semantics according to how much meaning every attack has for the computation of the extensions. It can be seen as an additional possibility to compare argumentation semantics. Amgoud and Vesic criticized in [2] that the notion of strong equivalence as introduced in [20] is too strong and has no practical application at all. We do agree that for logic-based argumentation systems no self-attacking arguments do exist, but if one uses a different formalism for the instantiation process, like the ASPIC+ system [21] or ASP (as proposed by Dung in [11]), self-attacking arguments can occur. Therefore, knowing about redundant attacks for specific semantics, and the classification of them in terms of succinctness, is useful and can make the evaluation easier. As redundant attacks have no influence, they can be omitted already during the instantiation process which can be a useful optimization step.

For future work, we will investigate the option to adapt the *cf2* semantics by replacing the naive semantics by stage semantics as a base function; recall Example 2.10 where we illustrated a certain undesired behavior of the *cf2* semantics for even cycles. Concerning the newly introduced succinctness property, we have already classified most of the argumentation semantics which are considered important nowadays. So far, *cf2* semantics is the only one which satisfies this property. As a next step an investigation of the resolution-based semantics [5] is planned. In particular, we claim that the resolution-based grounded semantics behaves here like the grounded semantics itself, which, as we already know from [20] and Theorem 5.8, violates the succinctness property. Finally, it also would be interesting to consider different levels of succinctness to have a more fine-grained classification of different semantics.

References

[1] Michael Abraham, Dov M. Gabbay, and Uri J. Schild. The Handling of Loops in Talmudic Logic, with Application to Odd and Even Loops in Argumentation. In David Rydeheard, Andrei Voronkov, and Margarita Korovina, editors, *Proceedings of Higher-Order Workshop on Automated Runntime Verification and Debugging* (*HOWARD 60*), 2011.

[2] Leila Amgoud and Srdjan Vesic. On the Equivalence of Logic-Based Argumentation Systems. In Salem Benferhat and John Grant, editors, *Proceedings of the 5th International Conference on Scalable Uncertainty Management* (*SUM 2011*), volume 6929 of *Lecture Notes in Computer Science*, pages 123–136. Springer, 2011.

[3] Pietro Baroni and Massimiliano Giacomin. Solving Semantic Problems with Odd-Length Cycles in Argumentation. In Thomas D. Nielsen and Nevin Lianwen Zhang, editors, *Proceedings of the 7th European Conference on Symbolic and Quantitative Approaches to Reasoning*

with Uncertainty, (*ECSQARU 2003*), volume 2711 of *Lecture Notes in Computer Science*, pages 440–451. Springer, 2003.

[4] Pietro Baroni and Massimiliano Giacomin. On Principle-based Evaluation of Extension-based Argumentation semantics. *Artif. Intell.*, 171(10-15):675–700, 2007.

[5] Pietro Baroni and Massimiliano Giacomin. Resolution-Based Argumentation Semantics. In Philippe Besnard, Sylvie Doutre, and Anthony Hunter, editors, *Proceedings of the 2nd Conference on Computational Models of Argument* (*COMMA 2008*), volume 172 of *Frontiers in Artificial Intelligence and Applications*, pages 25–36. IOS Press, 2008.

[6] Pietro Baroni and Massimiliano Giacomin. Semantics of Abstract Argument Systems. In Iyad Rahwan and Guillermo R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 25–44. Springer US, 2009.

[7] Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. SCC-Recursiveness: A General Schema for Argumentation Semantics. *Artif. Intell.*, 168(1-2):162–210, 2005.

[8] Martin Caminada. Semi-Stable Semantics. In Paul E. Dunne and Trevor J. M. Bench-Capon, editors, *Proceedings of the 1st Conference on Computational Models of Argument* (*COMMA 2006*), volume 144 of *Frontiers in Artificial Intelligence and Applications*, pages 121–130. IOS Press, 2006.

[9] Martin Caminada and Leila Amgoud. On the Evaluation of Argumentation Formalisms. *Artif. Intell.*, 171(5-6):286–310, 2007.

[10] Yannis Dimopoulos and Alberto Torres. Graph Theoretical Structures in Logic Programs and Default Theories. *Theor. Comput. Sci.*, 170(1–2):209–244, 1996.

[11] Phan Minh Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artif. Intell.*, 77(2):321–358, 1995.

[12] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Computing Ideal Sceptical Argumentation. *Artif. Intell.*, 171(10–15):642–674, 2007.

[13] Paul E. Dunne. Computational Properties of Argument Systems Satisfying Graph-Theoretic Constraints. *Artif. Intell.*, 171(10–15):701–729, 2007.

[14] Paul E. Dunne and Martin Caminada. Computational Complexity of Semi-Stable Semantics in Abstract Argumentation Frameworks. In Steffen Hölldobler, Carsten Lutz, and Heinrich Wansing, editors, *Proceedings of 11th European Conference on Logics in Artificial Intelligence* (*JELIA 2008*), volume 5293 of *LNCS*, pages 153–165. Springer, 2008.

[15] Wolfgang Dvorák and Stefan Woltran. Complexity of Semi-Stable and Stage Semantics in Argumentation Frameworks. *Inf. Process. Lett.*, 110(11):425–430, 2010.

[16] Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran. Answer-set Programming Encodings for Argumentation Frameworks. *Argument and Computation*, 1(2):144–177, 2010.

[17] Sarah Alice Gaggl and Stefan Woltran. cf2 Semantics Revisited. In Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Guillermo Ricardo Simari, editors, *Proceedings of the 3rd Conference on Computational Models of Argument (COMMA 2010)*, volume 216 of *Frontiers in Artificial Intelligence and Applications*, pages 243–254. IOS Press, 2010.

[18] Sarah Alice Gaggl and Stefan Woltran. Strong Equivalence for Argumentation Semantics Based on Conflict-Free Sets. In Weiru Liu, editor, *Proceedings of the 11th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2011)*, volume 6717 of *Lecture Notes in Computer Science*, pages 38–49. Springer, 2011.

[19] Juan Carlos Nieves, Mauricio Osorio, and Claudia Zepeda. Expressing Extension-Based Semantics Based on Stratified Minimal Models. In Hiroakira Ono, Makoto Kanazawa, and Ruy J. G. B. de Queiroz, editors, *Proceedings of the 16th International Workshop on Logic, Language, Information and Computation, (WoLLIC 2009)*, volume 5514 of *LNCS*, pages 305–319. Springer, 2009.

[20] Emilia Oikarinen and Stefan Woltran. Characterizing Strong Equivalence for Argumentation Frameworks. *Artif. Intell.*, 175(14-15):1985–2009, 2011.

[21] Henry Prakken. An Abstract Framework for Argumentation with Structured Arguments. *Argument and Computation*, 1(2):93–124, 2010.

[22] Robert Endre Tarjan. Depth-First Search and Linear Graph Algorithms. *SIAM J. Comput.*, 1(2):146–160, 1972.

[23] Bart Verheij. Two Approaches to Dialectical Argumentation: Admissible Sets and Argumentation Stages. In *Proceedings of the biannual International Conference on Formal and Applied Practical Reasoning (FAPR 1996) Workshop*, pages 357–368, 1996.