Minimal Strong Admissibility: a Complexity Analysis

Martin CAMINADA [a] and Paul E. DUNNE [b]
[a] *Cardiff University, Queen's Buildings, 5 The Parade, Cardiff CF24 2LQ, UK (CaminadaM@cardiff.ac.uk)*
[b] *University of Liverpool, Ashton Building, Liverpool L69 7ZF, UK (P.E.Dunne@liverpool.ac.uk)*

Abstract. The concept of strong admissibility plays an important role in some of the dialectical proof procedures that have been stated for grounded semantics. As the grounded extension is the (unique) biggest strongly admissible set, to show that an argument is in the grounded extension it suffices to show that it is in a strongly admissible set. We are interested in identifying a strongly admissible set that minimizes the number of steps needed in the associated dialectical proof procedure. In the current work, we look at the computational complexity of doing so.

Keywords. strong admissibility, computational complexity, explainable AI

1. Introduction

The concept of strong admissibility was first introduced in the work of Baroni and Giacomin [1] and has subsequently been studied by Caminada and Dunne [6,4]. Strong admissibility is particularly useful for showing that a particular argument is part of the grounded extension. As the grounded extension is the (unique) biggest strongly admissible set, showing membership of any strongly admissible set is sufficient to prove that the argument is in the grounded extension.

Alternatively, one could apply the concept of a strongly admissible labelling [6,4]. As the grounded labelling is the (unique) biggest strongly admissible labelling,[1] showing that an argument is labelled `in` by any strongly admissible labelling is sufficient to prove that the argument is labelled `in` by the grounded labelling (and therefore is an element of the grounded extension [5,11]).

As an argument can be labelled `in` by more than one strongly admissible labelling (or be an element of more than one strongly admissible set) the question then becomes which particular strongly admissible labelling to show in order to prove membership of the grounded extension. Although in principle any strongly admissible labelling that labels the argument `in` will do, it can have advantages to select a strongly admissible labelling that is *minimal*, especially when the aim is explainability.

[1]Biggest w.r.t. "$\sqsubseteq$" [12,4].

The concept of a strongly admissible labelling matters because it is at the basis of some of the proof procedures for grounded semantics [4], in particular of the Grounded Discussion Game [7]. The Grounded Discussion Game is a dialectical proof procedure with two players: the proponent and the opponent. The game is such that an argument is in the grounded extension iff it is possible for the proponent to win the discussion. The idea is that this discussion can serve as an explanation of why a particular argument should be accepted as being in the grounded extension. In such a case, the computer will assume the role of proponent and a human user will assume the role of opponent [3]. If an argument is in the grounded extension, the proponent can win the discussion by using a strongly admissible labelling as a roadmap [6]. In order to minimize the number of discussion steps (and hence save the user's time during the discussion) the strongly admissible labelling that is to be applied as a roadmap should have a minimal size[2] among all strongly admissible labellings that label the argument `in`. In the current paper we examine the computational complexity of verifying such a minimal strongly admissible labelling. In addition, we study the computational complexity of determining whether there is a strongly admissible labelling that labels a particular argument `in` and has a size of at most k.

This paper is structured as follows. First, in Section 2 we present some formal preliminaries regarding abstract argumentation and strong admissibility. Then, in Section 3 we present some results regarding the computational complexity of identifying strongly admissible labellings with bounded or minimal size. We round off in Section 4 with a discussion of the obtained results.

2. Preliminaries

For current purposes, we restrict ourselves to finite argumentation frameworks.

Definition 1. *An* argumentation framework *is a pair* (Ar, att) *where* Ar *is a finite set of entities, called arguments, whose internal structure can be left unspecified, and* att *is a binary relation on* Ar*. For any* $A, B \in Ar$ *we say that* A attacks B *iff* $(A, B) \in att$.

Definition 2. *Let* (Ar, att) *be an argumentation framework,* $A \in Ar$ *and* $\mathcal{A}rgs \subseteq Ar$*. We define* A^+ *as* $\{B \in Ar \mid A \text{ attacks } B\}$*,* A^- *as* $\{B \in Ar \mid B \text{ attacks } A\}$*,* $\mathcal{A}rgs^+$ *as* $\cup\{A^+ \mid A \in \mathcal{A}rgs\}$*, and* $\mathcal{A}rgs^-$ *as* $\cup\{A^- \mid A \in \mathcal{A}rgs\}$*.* $\mathcal{A}rgs$ *is said to be* conflict-free *iff* $\mathcal{A}rgs \cap \mathcal{A}rgs^+ = \emptyset$*.* $\mathcal{A}rgs$ *is said to* defend A *iff* $A^- \subseteq \mathcal{A}rgs^+$*. The characteristic function* $F : 2^{Ar} \rightarrow 2^{Ar}$ *is defined as* $F(\mathcal{A}rgs) = \{A \mid \mathcal{A}rgs \text{ defends } A\}$.

Definition 3. *Let* (Ar, att) *be an argumentation framework.* $\mathcal{A}rgs \subseteq Ar$ *is*

- *an admissible set iff* $\mathcal{A}rgs$ *is conflict-free and* $\mathcal{A}rgs \subseteq F(\mathcal{A}rgs)$
- *a complete extension iff* $\mathcal{A}rgs$ *is conflict-free and* $\mathcal{A}rgs = F(\mathcal{A}rgs)$
- *a grounded extension iff* $\mathcal{A}rgs$ *is the smallest (w.r.t.* $\subseteq$*) complete extension*
- *a preferred extension iff* $\mathcal{A}rgs$ *is a maximal (w.r.t.* $\subseteq$*) complete extension*

[2]We recall that the size of a labelling $\mathcal{L}ab$ is $|\mathtt{in}(\mathcal{L}ab) \cup \mathtt{out}(\mathcal{L}ab)|$.

The concept of strong admissibility was introduced by Baroni and Giacomin [1]. For current purposes we will apply the equivalent definition of Caminada [6,4].

Definition 4. *Let (Ar, att) be an argumentation framework. $\mathcal{A}rgs \subseteq Ar$ is* strongly admissible *iff every $A \in \mathcal{A}rgs$ is defended by some $\mathcal{A}rgs' \subseteq \mathcal{A}rgs \setminus \{A\}$ which in its turn is again strongly admissible.*

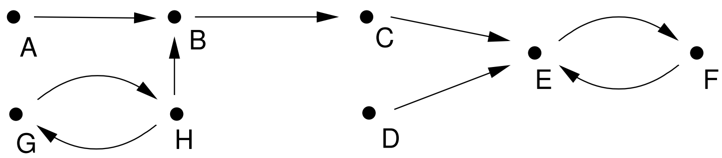

Figure 1. An example of an argumentation framework.

As an example (taken from [4]), in the argumentation framework of Figure 1 the strongly admissible sets are $\emptyset$, $\{A\}$, $\{A, C\}$, $\{A, C, F\}$, $\{D\}$, $\{A, D\}$, $\{A, C, D\}$, $\{D, F\}$, $\{A, D, F\}$ and $\{A, C, D, F\}$, the latter also being the grounded extension. The set $\{A, C, F\}$ is strongly admissible as A is defended by $\emptyset$, C is defended by $\{A\}$ and F is defended by $\{A, C\}$, each of which is a strongly admissible subset of $\{A, C, F\}$ not containing the argument it defends. Please notice that although the set $\{A, F\}$ defends argument C in $\{A, C, F\}$, it is in its turn not strongly admissible (unlike $\{A\}$). Hence the requirement in Definition 4 for $\mathcal{A}rgs'$ to be a *subset* of $\mathcal{A}rgs \setminus \{A\}$. We also observe that although $\{C, H\}$ is an admissible set, it is not a *strongly* admissible set, since no subset of $\{C, H\} \setminus \{H\}$ defends H.

It can be shown that each strongly admissible set is conflict-free and admissible [4]. The strongly admissible sets form a lattice, of which the empty set is the bottom element and the grounded extension is the top element [4].

The above definitions essentially follow the extension based approach as described in [13]. It is also possible to define the key argumentation concepts in terms of argument labellings [5,11].

Definition 5. *Let (Ar, att) be an argumentation framework. An* argument labelling *is a function $\mathcal{L}ab : Ar \to \{\texttt{in}, \texttt{out}, \texttt{undec}\}$. An argument labelling is called an* admissible labelling *iff for each $A \in Ar$ it holds that:*

- *if $\mathcal{L}ab(A) = \texttt{in}$ then for each B that attacks A it holds that $\mathcal{L}ab(B) = \texttt{out}$*
- *if $\mathcal{L}ab(A) = \texttt{out}$ then there exists a B that attacks A such that $\mathcal{L}ab(B) = \texttt{in}$*

$\mathcal{L}ab$ is called a complete labelling *iff it is an admissible labelling and for each $A \in Ar$ it also holds that:*

- *if $\mathcal{L}ab(A) = \texttt{undec}$ then there is a B that attacks A such that $\mathcal{L}ab(B) = \texttt{undec}$, and for each B that attacks A such that $\mathcal{L}ab(B) \neq \texttt{undec}$ it holds that $\mathcal{L}ab(B) = \texttt{out}$*

As a labelling is essentially a function, we sometimes write it as a set of pairs. Also, if $\mathcal{L}ab$ is a labelling, we write $\texttt{in}(\mathcal{L}ab)$ for $\{A \in Ar \mid \mathcal{L}ab(A) = \texttt{in}\}$, $\texttt{out}(\mathcal{L}ab)$ for $\{A \in Ar \mid \mathcal{L}ab(A) = \texttt{out}\}$ and $\texttt{undec}(\mathcal{L}ab)$ for $\{A \in Ar \mid \mathcal{L}ab(A) = \texttt{undec}\}$. As a labelling is also a partition of the arguments into sets of $\texttt{in}$-labelled arguments, $\texttt{out}$-labelled arguments and $\texttt{undec}$-labelled arguments, we sometimes write it as a triplet $(\texttt{in}(\mathcal{L}ab), \texttt{out}(\mathcal{L}ab), \texttt{undec}(\mathcal{L}ab))$.

Definition 6 ([12])**.** *Let $\mathcal{L}ab$ and $\mathcal{L}ab'$ be argument labellings of argumentation framework (Ar, att). We say that $\mathcal{L}ab \sqsubseteq \mathcal{L}ab'$ iff $\texttt{in}(\mathcal{L}ab) \subseteq \texttt{in}(\mathcal{L}ab')$ and $\texttt{out}(\mathcal{L}ab) \subseteq \texttt{out}(\mathcal{L}ab')$.*

Definition 7. *Let $\mathcal{L}ab$ be a complete labelling of argumentation framework (Ar, att). $\mathcal{L}ab$ is said to be*

- *a grounded labelling iff $\mathcal{L}ab$ is the (unique) smallest (w.r.t. $\sqsubseteq$) complete labelling*
- *a preferred labelling iff $\mathcal{L}ab$ is a maximal (w.r.t. $\sqsubseteq$) complete labelling*

The next step is to define a strongly admissible labelling. In order to do so, we first need to introduce the concept of a min-max numbering [4].

Definition 8. *Let $\mathcal{L}ab$ be an admissible labelling of argumentation framework (Ar, att). A* min-max numbering *is a total function $\mathcal{MM}_{\mathcal{L}ab} : \texttt{in}(\mathcal{L}ab) \cup \texttt{out}(\mathcal{L}ab) \rightarrow \mathbb{N} \cup \{\infty\}$ such that for each $A \in \texttt{in}(\mathcal{L}ab) \cup \texttt{out}(\mathcal{L}ab)$ it holds that:*

- *if $\mathcal{L}ab(A) = \texttt{in}$ then $\mathcal{MM}_{\mathcal{L}ab}(A) = max(\{\mathcal{MM}_{\mathcal{L}ab}(B) \mid B$ attacks A and $\mathcal{L}ab(B) = \texttt{out}\}) + 1$ (with $max(\emptyset)$ defined as 0)*
- *if $\mathcal{L}ab(A) = \texttt{out}$ then $\mathcal{MM}_{\mathcal{L}ab}(A) = min(\{\mathcal{MM}_{\mathcal{L}ab}(B) \mid B$ attacks A and $\mathcal{L}ab(B) = \texttt{in}\}) + 1$ (with $min(\emptyset)$ defined as ∞)*

It has been proved that every admissible labelling has a unique min-max numbering [4]. A strongly admissible labelling can then be defined as follows [4].

Definition 9. *A* strongly admissible labelling *is an admissible labelling whose min-max numbering yields natural numbers only (so no argument is numbered ∞).*

As an example (taken from [4]), consider again the argumentation framework of Figure 1. Here, the admissible labelling $\mathcal{L}ab_1 = (\{A, C, F, G\}, \{B, E, H\}, \{D\})$ has min-max numbering $\{(A : 1), (B : 2), (C : 3), (E : 4), (F : 5), (G : \infty), (H : \infty)\}$, which means that it is not strongly admissible. The admissible labelling $\mathcal{L}ab_2 = (\{A, C, D, F\}, \{B, E\}, \{G, H\})$ has min-max numbering $\{(A : 1), (B : 2), (C : 3), (D : 1), (E : 2), (F : 3)\}$, which means that it is strongly admissible.

The strongly admissible labellings also form a lattice, of which the all-$\texttt{undec}$ labelling is the bottom element and the grounded labelling is the top element [4].

A strongly admissible set is at the basis of the Grounded Discussion Game [7], which is a sound and complete dialectical proof procedure for proving that an argument is in the grounded extension. The game is played by two parties, called the proponent and the opponent, who each utter moves that contain arguments. The proponent starts by uttering what is called the main argument.The rules of the game are such that the main argument is in the grounded extension iff the proponent has a winning strategy for the game. The proponent is able to play such a winning strategy by basing his moves on a strongly admissible labelling and its associated min-max numbering. As the main argument can be labelled $\texttt{in}$

by several strongly admissible labellings, this raises the question of which strongly admissible labelling to choose. If the aim is to use the Grounded Discussion Game for purposes of explanation and human-computer interaction (as is suggested in [8]) one would like to choose a strongly admissible labelling that minimizes the required number of steps in the associated discussion. It has been observed [7] that such a strongly admissible labelling $\mathcal{L}ab$ should have a minimal size (that is, $|\mathtt{in}(\mathcal{L}ab) \cup \mathtt{out}(\mathcal{L}ab)|$ should be minimal) among all strongly admissible labellings that label the main argument `in`.

3. Computational Complexity

We will, generally, exploit the criteria specified in Definition 9 in order to validate that the labellings in the constructions are, indeed, strongly admissible labellings.

Formally, the bounded labelling problem is given as:

BOUNDED STRONG ADMISSIBLE LABELLING (BSAL)
Instance: An AF, $\mathcal{H} = (Ar, att)$, an argument $x \in Ar$ and a positive integer $k \in \mathbb{N}$.
Question: Is there a strongly admissible labelling, $\mathcal{L}ab$, of Ar for which

$$\mathcal{L}ab(x) = \mathtt{in} \text{ and } |\{\ y\ :\ \mathcal{L}ab(y) = \mathtt{in}\ \} \cup \{\ y\ :\ \mathcal{L}ab(y) = \mathtt{out}\}|\ \leq\ k\ ?$$

Theorem 1. BSAL *is* NP*–complete.*

Proof. We first note that BSAL $\in$ NP by virtue of the fact that for any strongly admissible labelling

$$\mathcal{L}ab\ :\ Ar \rightarrow \{\mathtt{in}, \mathtt{out}, \mathtt{undec}\}$$

its correctness may be checked in polynomial time (cf. [4]).

In order to show that BSAL is NP–hard we use a reduction from the well-known NP–complete problem of CNF satisfiability (CNF-SAT).

Given $\varphi(Z)$ a CNF formula over the propositional variables $Z = \{z_1, \ldots, z_n\}$ and having m clauses, $\{C_1, C_2 \ldots, C_m\}$ we form the AF, $\mathcal{H}_\varphi(Ar_\varphi, att_\varphi)$ with $|Ar_\varphi| = 3n + m + 1$ and arguments named

φ	
C_j	For each clause C_j and $1 \leq j \leq m$
D_i	For each variable z_i in Z
z_i	For each variable z_i in Z
$\neg z_i$	For each variable z_i in Z

The attacks in att_φ are

$< C_j, \varphi >$	For each $1 \leq j \leq m$
$< D_i, \varphi >$	For each $1 \leq i \leq n$
$< z_i, D_i >$	For each $1 \leq i \leq n$
$< \neg z_i, D_i >$	For each $1 \leq i \leq n$
$< z_i, C_j >$	if z_i is a literal in clause C_j of $\varphi(Z)$
$< \neg z_i, C_j >$	if $\neg z_i$ is a literal in clause C_j of $\varphi(Z)$

The instance of BSAL is formed as $< \mathcal{H}_\varphi,\ \varphi,\ 1+m+2n >$. Notice that as $\mathcal{H}_{varphi}$ is an acyclic AF, each admissible labelling is a strongly admissible labelling [9] and vice versa. We therefore only need to prove admissibility in order to show strong admissibility.

We claim this instance is accepted if and only if $\varphi(Z)$ is satisfiable.

First notice that **any** admissible labelling of $\mathcal{H}_\varphi$ in which $\mathcal{L}ab(\varphi) = \texttt{in}$ must be such that $|\{x\ :\ \mathcal{L}ab(x) = \texttt{undec}\}| \leq n$. In other words a labelling with minimal size must fix the status of at least $1+m+2n$ arguments. If $\mathcal{L}ab(\varphi) = \texttt{in}$ then we must have $\mathcal{L}ab(C_j) = \texttt{out}$ for every $1 \leq j \leq m$ and $\mathcal{L}ab(D_i) = \texttt{out}$ for every $1 \leq i \leq n$. In order to ensure the second of these we must have at least one of $\mathcal{L}ab(z_i) = \texttt{in}$ or $\mathcal{L}ab(\neg z_i) = \texttt{in}$. In total any strongly admissible labelling with $\mathcal{L}ab(\varphi) = \texttt{in}$ commits at least $1 + m + 2n$ arguments to a definite status (`in` or `out`).

Now suppose that $\varphi(Z)$ is satisfiable using some setting $\alpha = (a_1, a_2, \ldots, a_n)$ of its propositional variables. Choose the labelling, $\mathcal{L}ab_\alpha$ of Ar_φ for which

$$\mathcal{L}ab_\alpha(x) \quad = \quad = \begin{cases} \texttt{in} & \text{if } x = \varphi \\ \texttt{out} & \text{if } x \in \{C_1, \ldots, C_m\} \\ \texttt{out} & \text{if } x \in \{D_1, \ldots, D_n\} \\ \texttt{in} & \text{if } x = z_i \text{ and } a_i = \textbf{true} \\ \texttt{undec} & \text{if } x = z_i \text{ and } a_i = \textbf{false} \\ \texttt{in} & \text{if } x = \neg z_i \text{ and } a_i = \textbf{false} \\ \texttt{undec} & \text{if } x = \neg z_i \text{ and } a_i = \textbf{true} \end{cases}$$

It is not hard to see that this labelling satisfies the requirements needed to be an strongly admissible labelling: each $\{z_i, \neg z_i\}$ is unattacked and may be labelled as either `undec` or `in`; every D_i argument is correctly labelled `out` since it is attacked by an argument labelled `in` (i.e. z_i or $\neg z_i$); every C_j argument is, also, correctly labelled `out` as, since the labelling of $\{z_i,\ \neg z_i\ :\ 1 \leq i \leq n\}$ is determined by α (with $\varphi(\alpha) = \textbf{true}$) it follows that every C_j is attacked by some argument labelled `in` (since, in order for $\varphi(\alpha)$ to be **true**, every clause C_j must contain a literal which evaluates to **true** under α). Finally there are exactly n (the minimum possible) arguments labelled `undec`.

We conclude that if $\varphi(Z)$ is satisfiable then $< \mathcal{H}_\varphi,\ \varphi,\ 1+m+2n >$ is accepted as an instance of BSAL.

For the converse argument, suppose $< \mathcal{H}_\varphi,\ \varphi,\ 1+m+2n >$ is accepted as an instance of BSAL. Let $\mathcal{L}ab$ be the labelling of Ar_φ which witnesses this. That is to say, $\mathcal{L}ab(\varphi) = \texttt{in}$ and $|\{x\ :\ \mathcal{L}ab(x) \neq \texttt{undec}\}| = 1 + m + 2n$.

As we argued previously, from the fact that $\mathcal{L}ab(\varphi) = \texttt{in}$ we must have an additional $m + n$ arguments whose status is committed to being `out`: namely the $n + m$ clause arguments $\{\ C_j\ :\ 1 \leq j \leq m\} \cup \{\ D_i\ :\ 1 \leq i \leq n\}$. Furthermore for the labelling correctly to ensure $\mathcal{L}ab(D_i) = \texttt{out}$ we need either $\mathcal{L}ab(z_i) = \texttt{in}$ or $\mathcal{L}ab(\neg z_i) = \texttt{in}$. Now since we have assumed that $\mathcal{L}ab$ commits the status of at most $1 + m + 2n$ arguments and we have already determined how $1 + m + 2n$ must be set it must be the case that *exactly* one of $\{\ z_i, \neg z_i\}$ is set to `in` and the other to `undec`. Consider the setting, $\alpha_{\mathcal{L}ab}$ of the propositional variables:

$$\alpha_{\mathcal{L}ab}(z_i) \quad = \quad \begin{cases} \textbf{true} & \text{if } \mathcal{L}ab(z_i) = \texttt{in} \\ \textbf{false} & \text{if } \mathcal{L}ab(\neg z_i) = \texttt{in} \end{cases}$$

This assignment must satisfy $\varphi(Z)$: every clause argument, C_j, is correctly labelled out by $\mathcal{L}ab$ and, therefore, must be attacked by some z_i or $\neg z_i$ labelled in. In the assignment $\alpha_{\mathcal{L}ab}$ just described the corresponding setting of z_i as **true** ($\mathcal{L}ab(z_i) = \texttt{in}$) or **false** ($\mathcal{L}ab(\neg z_i) = \texttt{in}$) will lead to the clause C_j taking the value **true**.

We deduce that if $< \mathcal{H}_\varphi,\ \varphi,\ 1+m+2n >$ is accepted as an instance of BSAL then $\varphi(Z)$ is accepted as an instance of CNF-SAT. □

The decision problem BSAL is in essence an *existence* question: can we find a suitable labelling that commits the status of *at most* some number of arguments? A related question is that of *verifying* that a given labelling is indeed minimal. Formally this is the verification problem, MSAL:

MINIMAL STRONG ADMISSIBLE LABELLING (MSAL)
Instance: An AF, $\mathcal{H} = (Ar, att)$, an argument $x \in Ar$ and a strongly admissible labelling, $\mathcal{L}ab$ of Ar with which $\mathcal{L}ab(x) = \texttt{in}$.
Question: Does $\mathcal{L}ab$ have a minimal size? i.e. for any strongly admissible labelling, $\mathcal{L}ab'$, of Ar with $\mathcal{L}ab'(x) = \texttt{in}$, $|\{y : \mathcal{L}ab'(y) \neq \texttt{undec}\} \geq |\{y : \mathcal{L}ab(y) \neq \texttt{undec}\}$?

Theorem 2. MSAL *is co*NP*–complete.*

Proof. First notice that MSAL $\in$ coNP. Simply check every labelling $\mathcal{L}ab'$ of Ar and for any which describe a strongly admissible labelling with $\mathcal{L}ab'(x) = \texttt{in}$ confirm that $|\{y : \mathcal{L}ab'(y) \neq \texttt{undec}\}| \geq |\{y : \mathcal{L}ab(y) \neq \texttt{undec}\}$. This entire computation may be realized in coNP.

For coNP–hardness we use a reduction from CNF-UNSAT.

A key point in this reduction are that the instances of CNF-UNSAT are restricted to those having n propositional variables *and* exactly $m = 4n-1$ clauses.[3]

Given $\varphi(Z)$ a propositional formula over n variables and $4n - 1$ clauses $\{C_1, \ldots, C_{4n-1}\}$ the AF, $\mathcal{G}_\varphi$ consists of two parts:

1. The AF, $\mathcal{H}_\varphi$ from the proof of Theorem 1. Notice that this contains exactly $7n$ arguments: the literals $\{\ z_i,\ \neg z_i\ :\ 1 \leq i \leq n\}$; $4n-1$ clauses $\{C_j\ :\ 1 \leq j \leq 4n-1\}$; n clauses $\{D_i\ :\ 1 \leq i \leq n\}$ and φ.
2. The second section also uses the literal ($z_i,\ \neg z_i$) arguments from $\mathcal{H}_\varphi$ and an additional $4n + 1$ arguments:

$$\begin{array}{cc} \{\ b_i,\ \neg b_i\ :\ 1 \leq i \leq n\} & \{\ c_i\ :\ 1 \leq i \leq n\} \\ \{\ g_i\ :\ 1 \leq i \leq n\} & \pi \end{array}$$

In order to combine these structures two further arguments are introduced: ψ whose only attackers are φ and π; and θ whose only attacker is ψ.

The AF is completed by adding to those already in $\mathcal{H}_\varphi$ and the three attacks

[3] A standard "padding" argument such as that from [15, Thm. 2] easily shows this variant remains coNP–complete.

$$\{< \varphi, \psi >, < \pi, \psi >, < \psi, \theta >\}$$

the new attacks:

$$\begin{array}{lll} \{< z_i, b_i > \; : \; 1 \leq i \leq n\} & \quad & \{< \neg z_i, \neg b_i > \; : \; 1 \leq i \leq n\} \\ \{< b_i, c_i > \; : \; 1 \leq i \leq n\} & & \{< \neg b_i, c_i > \; : \; 1 \leq i \leq n\} \\ \{< c_i, g_i > \; : \; 1 \leq i \leq n\} & & \{< g_i, \pi > \; : \; 1 \leq i \leq n\} \end{array}$$

$\mathcal{G}_\varphi$ is illustrated in Figure 2. As $\mathcal{G}_\varphi$ is acyclic, it suffices to prove admissibility in order to show strong admissibility [9].

The labelling, $\mathcal{L}ab$, of which the minimal size is to be checked uses

$$\mathcal{L}ab(x) \quad = \quad \begin{cases} \texttt{in} & \text{if } x \in \{z_i, \ \neg z_i \; : \; 1 \leq i \leq n\} \\ \texttt{out} & \text{if } x \in \{b_i, \ \neg b_i \; : \; 1 \leq i \leq n\} \\ \texttt{in} & \text{if } x \in \{c_i \; : \; 1 \leq i \leq n\} \\ \texttt{out} & \text{if } x \in \{g_i \; : \; 1 \leq i \leq n\} \\ \texttt{in} & \text{if } x = \pi \\ \texttt{out} & \text{if } x = \psi \\ \texttt{in} & \text{if } x = \theta \\ \texttt{undec} & \text{otherwise} \end{cases}$$

We claim that $< \mathcal{G}_\varphi, \theta, \mathcal{L}ab >$ is accepted as an instance of MSAL if and only if $\varphi(Z_n)$ is unsatisfiable.

Suppose that $\varphi(Z)$ is in fact satisfiable using an assignment of propositional values $(a_1, \ldots, a_n)$. Notice that $\mathcal{L}ab$ has exactly $m + n + 1$ arguments labelled `undec` which given the conditions on m evaluates to $5n$. Consider the alternative labelling, $\mathcal{L}ab'$, in which

$$\mathcal{L}ab'(x) \quad = \quad \begin{cases} \texttt{in} & \text{if } \ x = z_i \text{ and } a_i = \textbf{true} \\ \texttt{in} & \text{if } \ x = \neg z_i \text{ and } a_i = \textbf{false} \\ \texttt{undec} & \text{if } \ x = z_i \text{ and } a_i = \textbf{false} \\ \texttt{undec} & \text{if } \ x = \neg z_i \text{ and } a_i = \textbf{true} \\ \texttt{undec} & \text{if } \ x \in \{b_i, \ \neg b_i \; : \; 1 \leq i \leq n\} \\ \texttt{undec} & \text{if } \ x \in \{c_i \; : \; 1 \leq i \leq n\} \\ \texttt{undec} & \text{if } \ x \in \{g_i \; : \; 1 \leq i \leq n\} \\ \texttt{undec} & \text{if } \ x = \pi \\ \texttt{out} & \text{if } \ x \in \{D_i \; : \; 1 \leq i \leq n\} \\ \texttt{out} & \text{if } \ x \in \{C_j \; : \; 1 \leq j \leq 4n - 1\} \\ \texttt{in} & \text{if } \ x = \varphi \\ \texttt{out} & \text{if } \ x = \psi \\ \texttt{in} & \text{if } \ x = \theta \end{cases}$$

The labelling, $\mathcal{L}ab'$ is easily checked to be a valid admissible labelling by virtue of the fact that $(a_1, \ldots, a_n)$ satisfies $\varphi(Z_n)$ every clause argument, C_j, can be labelled `out` since it is attacked by (at least one) z_i or $\neg z_i$ labelled `in`. Similarly each D_i is attacked by z_i labelled `in` or $\neg z_i$ labelled `in`. Finally since φ is attacked only by arguments labelled `out` it may be labelled `in` leading to $\mathcal{L}ab'(\psi) = \texttt{out}$ (the other attacker of ψ being `undec`) and $\mathcal{L}ab'(\theta) = \texttt{in}$. The number of `undec`

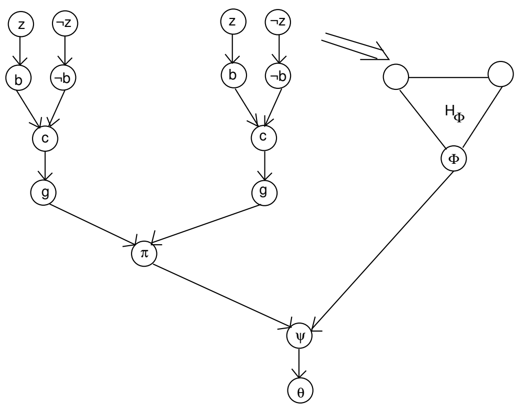

Figure 2. Construction of AF used to show MSAL is CONP–hard.

arguments is, however, more than those in $\mathcal{L}ab$ since $\mathcal{L}ab'$ labels n arguments (from $\{z_i,\ \neg z_i\}$) as `undec`, the $2n$ arguments in $\{\ b_i, \neg b_i\}$, the n arguments in $\{c_i : 1 \leq i \leq n\}$ and the n arguments in $\{g_i : 1 \leq i \leq n\}$. Finally π is also labelled `undec`. In total this gives $n + 2n + n + n + 1\ =\ 5n + 1$ so that,

$$|\{\ y\ :\ \mathcal{L}ab'(y) = \texttt{undec}\}|\ =\ 5n + 1\quad >\quad 5n\quad = |\{\ y\ :\ \mathcal{L}ab(y) = \texttt{undec}\}|$$

and the conclusion that if $\varphi(Z_n)$ is not accepted as an instance of CNF–UNSAT then $< \mathcal{G}_\varphi, \theta, \mathcal{L}ab >$ is not accepted as an instance of MSAL.

For the converse implication, suppose that $< \mathcal{G}_\varphi, \theta, \mathcal{L}ab >$ is rejected as an instance of MSAL.

In order for this to be the case we must have some admissible labelling, $\mathcal{L}ab'$, of $\mathcal{G}_\varphi$ in which $\mathcal{L}ab'(\theta) = \texttt{in}$ and $|\{y\ :\ \mathcal{L}ab'(y) = \texttt{undec}\}| > |\{y\ :\ \mathcal{L}ab(y) = \texttt{undec}\}$

It is not hard to see that any such labelling must use $\mathcal{L}ab'(\pi) = \texttt{undec}$ and $\mathcal{L}ab'(\varphi) = \texttt{in}$: in $\mathcal{L}ab$ every C_j and D_i argument together with φ are already `undec`; in order to ensure ψ can properly be labelled `out` at least one of π or φ must be labelled `in`. In order, however, properly to label π as `in` the status of **every** $\{z_i,\ \neg z_i\}$ has to be fixed.

Now in order for $\mathcal{L}ab'$ properly to label φ as `in` there are are two possibilities arising from the way in which $\{D_i\ :\ 1 \leq i \leq n\}$ may properly be labelled `out`.

Case 1: $\mathcal{L}ab'$ properly labels all C_j arguments as `out` through a labelling of $\{z_i, \neg z_i\}$ in which (at least) one z_i has $\mathcal{L}ab'(z_i) = \mathcal{L}ab'(\neg z_i) = \texttt{in}$.

Since at least one argument from each pair $\{z_i, \neg z_i\}$ must be committed to be `in` (in order properly to label D_i as `out`) a labelling, $\mathcal{L}ab'$ meeting the criteria in Case 1 contributes $n - 1$ `undec` (from $\{z_i,\ \neg z_i\}$); $2n$ (from $\{\ b_i, \neg b_i\}$); a further n ($\{c_i : 1 \leq i \leq n\}$); n more (from $\{g_i : 1 \leq i \leq n\}$) and the argument π. In total

$$(n - 1)\ +\ 2n\ + n\ +\ n\ +\ 1\ \ =\ \ 5n$$

Thus Case 1 (effectively using an invalid assignment to satisfy φ as a variable needs to be both **true** and **false**) leads to a labelling which is exactly the same size as $\mathcal{L}ab$: both have exacly $5n$ `undec` arguments.

Case 2: $\mathcal{L}ab'$ properly labels all C_j arguments as `out` through a labelling of $\{z_i, \neg z_i\}$ in which exactly one of $\mathcal{L}ab'(z_i) = \texttt{in}$ or $\mathcal{L}ab'(\neg z_i) = \texttt{in}$ holds.

Now this case has n (z arguments) together with $4n + 1$ other arguments ($\{b_i, \neg b_i\}$, $\{c_i\}$, $\{g_i\}$, π) whose status is `undec`, leading to $n{+}2n{+}n{+}n{+}1 \; = \; 5n{+}1$ undecided arguments and a smaller number of committed arguments than $\mathcal{L}ab$. Consider, however, the assignment of propositional $(a_1, a_2, \ldots, a_n)$ values to Z formed through

$$a_i \quad = \quad \begin{cases} \textbf{true} \text{ if } \mathcal{L}ab'(z_i) = \texttt{in} \text{ and } \mathcal{L}ab'(\neg z_i) = \texttt{undec} \\ \textbf{false} \text{ if } \mathcal{L}ab'(z_i) = \texttt{undec} \text{ and } \mathcal{L}ab'(\neg z_i) = \texttt{in} \end{cases}$$

This assignment guarantees that every clause C_j of $\varphi(Z)$ will have at least one literal which evaluates to **true** (since the corresponding C_j argument is correctly labelledl `out` by virtue of being attacked by a literal labelled `in`).

In total $(a_1, a_2, \ldots, a_n)$ is a setting of Z in which every clause of φ contains a **true** literal, i.e. $(a_1, a_2, \ldots, a_n)$ witnesses that $\varphi(Z)$ would be rejected as an instance of CNF−UNSAT.

We deduce that if $< \mathcal{G}_\varphi, \theta, \mathcal{L}ab >$ is rejected as an instance of MSAL then $\varphi(Z)$ is rejected as an instance of CNF−UNSAT.

In total, $< \mathcal{G}_\varphi, \theta, \mathcal{L}ab >$ describes an admissible labelling with minimal size of θ as `in` if and only if φ is unsatisfiable. $\square$

4. Discussion

The concept of a strong admissibility is related to grounded semantics in a similar way as the concept of admissibility is related to preferred semantics. In order to prove that an argument is in the grounded extension, we do not have to construct the entire grounded extension. Instead, it is sufficient to construct a strongly admissible set containing it. Similarly, in order to prove that an argument is in a preferred extension, we do not have to construct the entire preferred extension. Instead, it is sufficient to construct an admissible set containing it.

In essence, constructing an admissible set is what is being done by the Preferred Discussion Game [10]. The rules of this game are such that an argument is in an admissible set (and therefore in a preferred extension) if the proponent has a winning strategy for this game. Such a winning strategy can be derived using an admissible set $\mathcal{A}rgs$ that contains the argument A in question. When doing so, the resulting game will have a number of moves that is no greater than $2 \cdot |\mathcal{A}rgs^{-}| + 1$. It has been shown [10] that in order to minimize the number of moves required in the Preferred Discussion Game, one needs to obtain an admissible set $\mathcal{A}rgs$ that contains A and where $|\mathcal{A}rgs^{-}|$ is minimal among all the admissible sets that contain A.

The desire to minimize $|\mathcal{A}rgs^{-}|$ leads to two relevant decision problems: that of *verification* where given an AF (Ar, att) and a set $\mathcal{A}rgs$ that contains argument A it is asked if $\mathcal{A}rgs$ is an admissible set where $|\mathcal{A}rgs^{-}|$ is minimal among all admissible sets containing A; and the *existence* where given an AF (Ar, att), an argument A and an integer k it is asked if there is an admissible set $\mathcal{A}rgs$ that contains A with $|\mathcal{A}rgs^{-}| \leq k$.

It was found that the verification problem is coNP–complete, and the existence problem is NP–complete [10].

Table 1 provides an overview of how the main results of the current paper (Theorem 1, Theorem 2) compare with the status of similar problems with respect to standard Dung-style admissibility.

Table 1. Admissibility vs. Strong Admissibility

Problem	Complexity (ADM)	Complexity (Strong ADM)
Verification	Polynomial	Polynomial
Acceptability	NP–complete	Polynomial
Minimal Labelling (existence)	NP–complete	NP–complete
Minimal Labelling (verification)	coNP–complete	coNP–complete

With the exception of (credulous) acceptability these have similar complexity. The discrepancy that acceptability is NP–complete (standard Dung admissibility) whereas the analogous decision problem for strong admissibility is polynomial time decidable, arises from the fact that there is a unique maximal (w.r.t $\subseteq$) strongly admissible set, namely the grounded extension. Thus a simple test as to whether x is contained in a strongly admissible set is just to check if x is in the grounded extension.

It is also worth noting the differences between the reductions to establish intractability as given for admissibility (from [10]) and the constructions in Theorem 1, Theorem 2 for the analogous strong admissibility problems. All four proofs turn on variations of the standard translation of CNF-SAT, see e.g [16, Defn. 5.1, p. 91]. In both [10, Theorem 6.6] (verification of labelling minimality) and [10, Theorem 6.7] (existence of labelling with given size) the constructions used cyclic AFs whose grounded extension is empty. For the cases considered in Theorems 1, 2 we need to have AFs with a non-empty grounded extension. The constructions used, however, go one step further as summarized in the following.

Theorem 3.

a. BSAL *is* NP*–complete if instances are restricted to acyclic frameworks.*
b. MSAL *is* coNP*–complete if instances are restricted to acyclic frameworks.*

Proof. Immediate from the proofs of Theorem 1 and Theorem 2. □

It is worth noting that while there are a very small number of intractability results involving acyclic AFs (e.g. [14, Theorem 23] with binary tree forms) typically these rely on developments of standard Dung frameworks, e.g. the result from [14] exploits properties of value–based argumentation from [2]).

The research of the current paper fits into our long-term research agenda of using argumentation theory to provide explainable formal inference. In our view, it is not enough for a knowledge-based system to simply provide an answer regarding what to do or what to believe. There should also be a way for this answer to be explained. One way of doing so is by means of (formal) discussion [8]. Here, the idea is that the knowledge-based system should provide the argument that is at the basis of its advice. The user is then allowed to raise objections (counter-arguments) which the system then replies to (using counter-counter-arguments), etc. In general, we would like such a discussion to be (1) sound and complete for the underlying argumentation semantics, (2) not be unnecessarily long, and (3) be close enough to human discussion in order to be perceived as natural and convincing.

As for point (1), sound and complete discussion games have been identified for grounded, preferred, stable and ideal semantics [8]. As for point (2), this is what we studied in the current paper, as well as in [10]. As for point (3), this is something that we are aiming to report on in future work.

References

[1] P. Baroni and M. Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence*, 171(10-15):675–700, 2007.

[2] T. J. M. Bench-Capon. Persuasion in Practical Argument Using Value-based Argumentation Frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

[3] R. Booth, M.W.A. Caminada, and B. Marshall. DISCO: A web-based implementation of discussion games for grounded and preferred semantics. In *Proceedings of COMMA 2018*, pages 453–454, 2018.

[4] M. W. A. Caminada and P. E. Dunne. Strong admissibility revised: theory and applications. *Argument & Computation*, 10:277–300, 2019.

[5] M.W.A. Caminada. On the issue of reinstatement in argumentation. In M. Fischer, W. van der Hoek, B. Konev, and A. Lisitsa, editors, *Logics in Artificial Intelligence; 10th European Conference, JELIA 2006*, pages 111–123. Springer, 2006. LNAI 4160.

[6] M.W.A. Caminada. Strong admissibility revisited. In S. Parsons, N. Oren, C. Reed, and F. Cerutti, editors, *Computational Models of Argument; Proceedings of COMMA 2014*, pages 197–208. IOS Press, 2014.

[7] M.W.A. Caminada. A discussion game for grounded semantics. In E. Black, S. Modgil, and N. Oren, editors, *Theory and Applications of Formal Argumentation (proceedings TAFA 2015)*, pages 59–73. Springer, 2015.

[8] M.W.A. Caminada. Argumentation semantics as formal discussion. In *Handbook of Formal Argumentation*, volume 1, pages 487–518. College Publications, 2018.

[9] M.W.A. Caminada. Strong admissibility in acyclic argumentation frameworks. Technical report, Cardiff University, 2020.

[10] M.W.A. Caminada, W. Dvořák, and S. Vesic. Preferred semantics as socratic discussion. *Journal of Logic and Computation*, 26:1257–1292, 2014.

[11] M.W.A. Caminada and D.M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009. Special issue: new ideas in argumentation theory.

[12] M.W.A. Caminada and G. Pigozzi. On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22(1):64–102, 2011.

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

[14] P. E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artificial Intelligence*, 171(10):701 – 729, 2007.

[15] P. E. Dunne, A. Gibbons, and M. Zito. Complexity-theoretic models of phase transitions in search problems. *Theoretical Computer Science*, 249(2):243 – 263, 2000.

[16] P. E. Dunne and M. J. Wooldridge. Complexity of abstract argumentation. In G. Simari and I. Rahwan, editors, *Argumentation in Artificial Intelligence*, pages 85–104. Springer US, Boston, MA, 2009.