Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220158

On the Complexity of Determining Defeat Relations Consistent with Abstract Argumentation Semantics

Jack MUMFORD [a,b], Isabel SASSOON [c], Elizabeth BLACK [b] and Simon PARSONS [d]

[a] *Department of Computer Science, University of Liverpool, UK*
[b] *Department of Informatics, King's College London, UK*
[c] *Department of Computer Science, Brunel University London, UK*
[d] *School of Computer Science, University of Lincoln, UK*

Abstract. Typically in abstract argumentation, one starts with arguments and a defeat relation, and applies some semantics in order to determine the acceptability status of the arguments. We consider the converse case where we have knowledge of the acceptability status of arguments and want to identify a defeat relation that is consistent with the known acceptability data – the σ-consistency problem. Focusing on complete semantics as underpinning the majority of the major semantic types, we show that the complexity of determining a defeat relation that is consistent with some set of acceptability data is highly dependent on how the data is labelled. The extension-based 2-valued σ-consistency problem for complete semantics is revealed as NP-complete, whereas the labelling-based 3-valued σ-consistency problem is solvable within polynomial time. We then present an informal discussion on application to grounded, stable, and preferred semantics.

Keywords. Abstract argumentation, Complexity analysis, σ-consistency

1. Introduction

The typical argumentation problem takes arguments, a defeat relation, and a labelling semantics as input and produces argument acceptability labellings as output. We instead examine a converse argumentation problem, which takes arguments, labelling semantics, and argument labellings as input and produces a defeat relation as output.

Argumentation is firmly established as a subject of importance for researchers interested in symbolic representations of knowledge and defeasible reasoning [5]. Argumentation frameworks (AFs) [10] offer a graph-based approach that determines logically consistent positions through the interaction of arguments solely through a defeat relation. Determining the presence, or absence, of defeats between arguments is fundamental to the construction of any AF and is the problem that concerns this paper. Given a set of arguments, and data on which arguments are acceptable, we examine the complexity of establishing a set of defeats that is consistent with the data.

In certain contexts a defeat relation between arguments can be reliably and efficiently inferred from the structure and content of the arguments themselves. However, if the structure and/or the content of the arguments cannot be relied upon, then we become more dependent upon the remaining two components of an AF: the labelling semantics, and the argument labellings. For example, enthymemes – arguments with missing structure – pose problems for inferring a defeat relation. Suppose if some argument a defeats some argument b via an undermining attack on an unstated implicit premise of b, then how is one to recognise the defeat exists? Perhaps knowledge of the contextual content of the arguments fills in the gap of the missing premise, but perhaps not. However, if we possess an argument labelling where a is accepted and b is rejected, then the defeat may be inferred from this information alone.

In this paper we shall examine the σ-consistency problem, whereby we seek a solution defeat relation that is wholly consistent with input acceptability data for a given semantics. It is not assumed that the input data are exhaustive; other labellings that are not present in the data set may also be consistent with the solution defeat relation. Moreover, the problem is general in that it accepts partial labellings where not every argument in the overall argument set need be represented in a given labelling. That is, an arbitrary labelling in the data set may pertain to a subgraph of arguments S where $S \subset A$, or it may pertain to all of A.

Example 1. *Legal domains are commonly modelled using stereotypical patterns of facts known as factors, following the CATO approach for legal case-based reasoning [1]. A case is decided by weighing up the balance of the factors from case precedents. This process could be modelled by the creation of a representative AF model in accordance with the σ-consistency problem. For example, in CATO's domain, US Trade Secret's Law, the two factors related to the issue of confidentiality were F1 (DisclosureInNegotiations) and F21 (KnewInfoConfidential). There is nothing explicit within the structure or content of the factors that definitively reveals how the factors are appropriately balanced. However, now suppose that we have a case history such that whenever F1 and F21 are argued (indeed F1 and/or F21 may not be argued in a case, but recall that the σ-consistency problem accepts partial labellings), we find F1 is rejected (labelled* OUT*) and F21 is accepted (labelled* IN*), we have a way of establishing a defeat relation where $(F21, F1) \in R$, that would be consistent with the precedents, thus presenting a method of balancing the factors. Conversely, if we then were presented with a new case in which both factors were accepted (labelled* IN*) then no defeat relation exists that would be consistent with the amended case history. But even a failure to produce a solution serves a purpose; indicating that the existing factor-based model of the domain may be insufficient and require broadening or narrowing. Note that in the actual CATO analysis such a converse case never arose and $(F21, F1) \in R$ was the appropriate conclusion.*

Whilst many semantics are available for investigation, our primary attention is placed on complete semantics due to the superset relation it forms with respect to the traditional major semantics as well as the intuition of it representing the range of reasonable positions that are deterministic consequences of the defeat relation (which admissibility semantics does not guarantee). We will show that the complexity of a σ-consistency problem with complete semantics (note from this point onward we shall refer to a 'σ-consistency with complete semantics' simply as a 'σ-consistency problem') depends significantly on the type of input acceptability data, where the extension-based 2-valued problem (arguments labelled IN or NOTIN) is **NP**-complete and the 3-valued problem (arguments labelled IN, UNDEC or OUT) can be solved within polynomial, specifically multi-variate quadratic, time complexity (although both are solvable within quadratic space complexity). We believe this paper offers the following contributions:

- Introduces and defines the σ-consistency problem, which asks if a solution defeat relation exists that is consistent with a set of argument acceptability labellings.
- Proves that the 2-valued σ-consistency problem is **NP**-complete and show its space complexity to be within $O(n^2)$, where n is the number of arguments.
- Proves the 3-valued σ-consistency problem is solvable by algorithms with a time complexity within $O(n^2|T|)$, where $|T|$ is the number of labellings in the input, and space complexity within $O(n^2)$.
- Provides informal complexity results for stable, preferred, and grounded semantics.
- Provides insight into the scaleability of algorithms designed to determine a defeat relation from argument acceptability data, suggesting a preference for sourcing 3-valued data to reduce the risk of solving high complexity problems.

2. Background

We draw upon the concept of argumentation underpinned by Dung's seminal paper [10] and the reinstatement principles advocated by Caminada [6] in developing our complexity proofs for the 2-valued and 3-valued σ-consistency problems.

An *argumentation framework* is a pair $\mathrm{AF} = \langle A, R \rangle$ where A is a set of arguments, and R is a binary relation on A, i.e. $R \subseteq A \times A$. We apply the notation for defeat from [6]: Let $a, b \in A$, we define a^+ as $\{b | a \text{ defeats } b\}$ and a^- as $\{b | b \text{ defeats } a\}$. A set S of arguments is said to be *conflict-free* if there are no arguments a and b in S such that $a \in b^-$. A set of arguments S defeats an argument a *iff* $\exists b \in S : b \in a^-$. An argument $a \in A$ is *acceptable* with respect to a set S *iff* for each argument $b \in A$: if $b \in a^-$ then $\exists c \in S : b \in c^+$. A conflict-free set of arguments S is *admissible iff* each argument in S is acceptable with respect to S. An admissible set S of arguments is a *complete extension iff* each argument that is acceptable with respect to S belongs to S. In a 2-valued complete labelling, all arguments in S are labelled IN, and all arguments in $A \setminus S$ are labelled NOTIN.

Def. 1 prescribes the implementation of the reinstatement labelling process to produce a 3-valued complete labelling, which coincides with complete extension semantics [7]. The formulation has been adopted from [24] with some notation changed in order to maintain consistency throughout this paper.

Definition 1. (3-valued complete labelling) *Let* $L : A \to \{\text{IN}, \text{UNDEC}, \text{OUT}\}$ *be a labelling of argumentation framework* (A, R)*. We say that L is a complete labelling iff for each argument* $b \in A$ *it holds that:*

1. $(L(b) = \text{IN}) \equiv (\forall a \in A : (a \in b^-) \Rightarrow (L(a) = \text{OUT}))$*;*
2. $(L(b) = \text{OUT}) \equiv (\exists a \in A : (a \in b^-) \wedge (L(a) = \text{IN}))$*; and*
3. $(L(b) = \text{UNDEC}) \equiv ((\exists a \in A : (a \in b^-) \wedge (L(a) = \text{UNDEC})) \wedge (\forall c \in A : (c \in b^-) \Rightarrow (L(c) \neq \text{IN}))$.

A signature $\Sigma_\sigma((A,R))$ of a semantics σ, is the collection of all possible sets of labellings from all subgraphs of $\mathrm{AF} = \langle A, R \rangle$ under semantics σ [11]. We use the concept of a signature for both the 2-valued and 3-valued σ-consistency problems, since any solution defeat relation R would require all input acceptability labellings to be in $\Sigma_\sigma((A,R))$.

Definition 2. (Signatures of argumentation semantics, adapted – with notation changes – from Dunne et al. [11]) *Let $\sigma((S,R))$ be the set of possible labellings given arguments S, defeat relation R, and argumentation semantics σ. We can express the set of all possible labellings over a set of input AFs as $\Sigma_\sigma((A,R))$ such that:*

$$\Sigma_\sigma((A,R)) = \{\sigma((S,R))|(S,R) \text{ is an AF and } S \subseteq A\}$$

3. Complexity of 2-valued σ-consistency

In order to formalise the 2-valued σ-consistency problem we must adapt the properties governing complete extension semantics, rendering explicit the 2-valued labelling form that the semantics prescribes: for each extension, an argument is labelled IN *iff* it is within the extension, and NOTIN *iff* it is outside the extension. Def. 3 concisely presents complete extension semantics according to the 2-valued labelling form. We then further adapt this form to produce our complexity results. We add the OFF label to represent those arguments that are unlabelled in a particular reinstatement labelling. The OFF label is essential for when we formalise the 2-valued σ-consistency problem, since it allows us to include partial labellings representing subgraphs of the wider AF.

Definition 3. (2-valued complete labelling) *Let A be a set of arguments; let $L : A \to \{\text{IN}, \text{NOTIN}, \text{OFF}\}$ be a labelling; let $R \subseteq A \times A$ be a defeat relation. $L \in \sigma((S,R))$, where $S = A \setminus X$, and where X is the set of arguments labelled* OFF*, iff each of the following conditions hold $\forall b \in S$:*

1. *$L(b) = \text{IN} \iff \forall a \in S : ((L(a) = \text{IN} \implies (a,b) \notin R) \wedge (L(a) = \text{NOTIN} \implies (a,b) \notin R \vee \exists c \in S : (L(c) = \text{IN} \wedge (c,a) \in R)))$; and*
2. *$L(b) = \text{NOTIN} \iff \exists a \in S : ((L(a) = \text{IN} \wedge (a,b) \in R) \vee (L(a) = \text{NOTIN} \wedge (a,b) \in R \wedge \forall c \in S : (L(c) = \text{IN} \implies (c,a) \notin R)))$.*

Def. 4 formally presents the 2-valued σ-consistency problem of finding a solution defeat relation set consistent with 2-valued input argument acceptability data. From this definition we produce the principle complexity result in Theorem 2, that the σ-consistency problem is **NP**-complete.

Definition 4. (2-valued σ-consistency problem) *Given a set of arguments A, a semantics σ, and a set of labellings T such that for each $L \in T$, $L : A \to \{\text{IN}, \text{NOTIN}, \text{OFF}\}$, is there a defeat relation $R \subseteq A \times A$ consistent with T such that $\forall L : (L \in \sigma((S,R)) \in \Sigma_\sigma((A,R)))$, where $S = A \setminus X$, and where X is the set of arguments labelled* OFF*?*

It should be obvious that in general one should not assume there exists just one solution defeat relation for an arbitrary 2-valued σ-consistency problem (e.g. an input labelling of two arguments where one argument is labelled IN, and the other NOTIN has more than one defeat relation that could produce this labelling). Before we move on to the full complexity proof, we make the intermediate observation in Theorem 1 of the symmetry of defeat inconsistency – that for complete semantics if some defeat $(a,b) \in R$ is inconsistent with some labelling then (b,a) would also be inconsistent with the labelling.

Theorem 1. (Bidirectional defeat inconsistency for 2-valued labellings) *Assuming complete argumentation semantics, for any set of labellings* $T : \forall L \in T,\ L : A \to \{\text{IN}, \text{NOTIN}, \text{OFF}\}$*, we have* $\exists a,b \in A : ((a,b) \in R \implies L \notin \sigma((S,R)) \iff (b,a) \in R \implies L \notin \sigma((S,R)))$*, where* $S = A \setminus X$*, and* X *is the set of arguments labelled* OFF.

Proof. Suppose we have σ as complete semantics, a and b are arbitrary arguments in A, and $S = A \setminus X$, and where X is the set of arguments labelled OFF.
(*Forward implication*) Suppose $(a,b) \in R \implies L \notin \sigma((S,R))$. From Def. 3 we draw expressions of the form $L(a) \wedge L(b) \implies (a,b) \notin R$. We have three cases:
Case 1. $\forall a,b \in S : (L(b) = \text{IN} \wedge L(a) = \text{IN} \implies (a,b) \notin R)$. Suppose $(b,a) \in R$, then by the first condition of Def. 3 $L(a) = \text{IN} \implies (b,a) \notin R$. Contradiction!
Case 2. $\forall a,b \in S : ((L(b) = \text{IN} \wedge L(a) = \text{NOTIN} \implies (a,b) \notin R) \implies \nexists c \in S : (L(c) = \text{IN} \wedge (c,a) \in R))$. Suppose $(b,a) \in R$, then $\exists c \in S : (L(c) = \text{IN} \wedge (c,a) \in R)$. Contradiction!
Case 3. $\forall a,b \in S : ((L(b) = \text{NOTIN} \wedge L(a) = \text{IN} \implies (a,b) \notin R) \implies \exists c \in S : (L(c) = \text{NOTIN} \wedge (b,c)) \wedge \forall d \in (S \setminus b) : (\forall e \in S : (L(e) = \text{IN} \implies (e,d) \notin R) \wedge (d,c) \notin R))$. Suppose $(b,a) \in R$, then by the first condition of Def. 3 $\exists f \in S : (L(f) = \text{IN} \wedge (f,b) \in R)$. Contradiction!
Therefore if $(a,b) \in R \implies L \notin \sigma((S,R))$ then $(b,a) \in R \implies L \notin \sigma((S,R))$.
(*Backward implication*) Since a and b are arbitrary, the backward implication immediately follows from the permutation $a \leftrightarrow b$ in the forward implication.
Therefore if $(b,a) \in R \implies L \notin \sigma((S,R))$ then $(a,b) \in R \implies L \notin \sigma((S,R))$. $\square$

We shall see later in the paper that the symmetry of inconsistency also holds for the 3-valued σ-consistency problem. Hence the symmetry of defeat inconsistency with acceptability data underpins any σ-consistency problem based on complete semantics. It is essential to understand that Theorem 1 applies to asymmetric as well as symmetric frameworks, but simply outlines that if an arbitrary defeat (a,b) is incompatible with a labelling set, then (b,a) will also be incompatible.

Theorem 2. (2-valued σ-consistency problem as **NP**-complete) *The 2-valued σ-consistency problem as defined in Def. 4 is* **NP***-complete.*

Proof. Suppose there is a set of arguments A, labellings T, and complete semantics σ.
It is easy to show that the 2-valued σ-consistency problem is in **NP**. Given a solution defeat relation R, from [8], the verification problem of confirming a given labelling L is consistent with R and σ, is solvable in polynomial-time. Since T is a fixed set of labellings, then the verification problem of confirming each labelling $L \in T$ is consistent with R and σ, is clearly also solvable within polynomial-time. Therefore the 2-valued σ-consistency problem is within **NP**.
We must show that the **NP**-complete Monotone 3SAT problem [14] can be reduced by function r to the 2-valued σ-consistency problem in polynomial-time. An arbitrary instance of a Monotone 3SAT problem consists of a set of clauses C, where we require that C is satisfied if and only if $\forall L_j \in r(C) : (L_j \in \sigma((S,R)))$, where $S = A \setminus X$, and where X is the set of arguments labelled OFF.
Suppose an arbitrary Monotone 3SAT problem and some arguments $\omega, \beta \in A$, where $\omega \neq \beta$. Suppose an arbitrary clause $c_i \in C$, then $a_j \in c_i \iff (a_j, \omega) \in R$ and $\neg a_j \in c_i \iff (a_j, \omega) \notin R$, where a_j is arbitrary. There exist two cases for an arbitrary clause c_i in an Monotone 3SAT problem; suppose we have r such that:

Case 1. $c_i = (a_i \vee b_i \vee c_i)$ and $r(c_i) = (S, L_{1i}, L_{2i})$ where $S = \{a_i, b_i, c_i, \omega\}$, $L_{1i} = \{L(a_i) = \text{NOTIN}, L(b_i) = \text{NOTIN}, L(c_i) = \text{NOTIN}, L(\omega) = \text{NOTIN}\}$, and $L_{2i} = \{L(\omega) = \text{IN}\}$, and a_i, b_i, c_i are arbitrary.
Case 2. $c_i = (\neg a_i \vee \neg b_i \vee \neg c_i)$ and $r(c_i) = (S, L_{3i}, L_{4i})$, where $S = \{a_i, b_i, c_i, \omega, \beta\}$ $L_{3i} = \{L(a_i) = \text{NOTIN}, L(b_i) = \text{NOTIN}, L(c_i) = \text{NOTIN}, L(\omega) = \text{IN}, L(\beta) = \text{NOTIN}\}$, and $L_{4i} = \{L(\omega) = \text{IN}, L(\beta) = \text{IN}\}$, and a_i, b_i, c_i are arbitrary.
We show the forward and backward implications hold for the two possible cases for arbitrary c_i via equivalence. In the reduction we use the shorthand (a,b) to indicate $(a,b) \in R$, and $\neg(a,b)$ to indicate $(a,b) \notin R$.
(*Case 1*) Suppose arbitrary clause $c_k = (a_i \vee b_i \vee c_i)$, from Def. 3 we have $r(c_k) \implies \bigwedge_{y \in S} (\bigvee_{x \in S} (x,y)) \wedge \neg(\omega, \omega)$. Since a_i, b_i and c_i are always labelled NOTIN in every labelling $L_j \in r(C)$, they are all free to defeat themselves and one another. Hence we are left to satisfy $r(c_k) \implies (a_i, \omega) \vee (b_i, \omega) \vee (c_i, \omega) \equiv c_k$.
(*Case 2*) Suppose arbitrary clause $c_m = (\neg a_i \vee \neg b_i \vee \neg c_i)$, from Def. 3 we have $r(c_m) \implies \bigwedge_{y \in S \setminus \omega} ((\omega, y) \vee \bigvee_{x \in S \setminus \omega} ((x,y) \wedge \neg(\omega, x))) \wedge \neg(\omega, \omega) \wedge \bigwedge_{z \in S \setminus \omega} ((z, \omega) \implies (\omega, z)) \wedge \neg(\omega, \omega) \wedge \neg(\omega, \beta) \wedge \neg(\beta, \omega) \wedge \neg(\beta, \beta)$. As outlined in the previous case, a_i, b_i and c_i are free to defeat themselves and one another, and this is also extended to β. Hence we are left to satisfy $r(c_m) \implies \bigvee_{x \in \{a_i, b_i, c_i\}} (\neg(\omega, x)) \wedge \bigwedge_{y \in \{a_i, b_i, c_i\}} ((y, \omega) \implies (\omega, y))$, which by modus tollens further reduces to $r(c_m) \implies \neg(a_i, \omega) \vee \neg(b_i, \omega) \vee \neg(c_i, \omega) \equiv c_m$.
It is clear that the reduction produces a permutation where $((a_i, \omega) \in R) \equiv (a_i = \top)$, $((a_i, \omega) \notin R) \equiv (\neg a_i = \top)$, $((b_i, \omega) \in R) \equiv (b_i = \top)$, $((b_i, \omega) \notin R) \equiv (\neg b_i = \top)$, $((c_i, \omega) \in R) \equiv (c_i = \top)$ and $((c_i, \omega) \notin R) \equiv (\neg c_i = \top)$. Therefore for both cases of arbitrary clause $c_i \in C$, the reduction simply produces a permutation of the monotone 3SAT problem and consequently C is satisfied if and only if $\forall L_j \in r(C) : (L_j \in \sigma((S,R)))$.
It is clear that $r : c_i \mapsto \{S, T\}$, where $S \subseteq \{a_i, b_i, c_i, \omega, \beta\}$ and $T \subseteq \{L_{1i}, L_{2i}, L_{3i}, L_{4i}\}$, requires constant number of operations to produce the elements in S and T for any arbitrary clause $c_i \in C$. Therefore for k clauses the reduction requires $O(k)$ operations, which is within polynomial-time.
Therefore the 2-valued σ-consistency problem is **NP**-complete. □

Deriving the space complexity result is a much simpler affair. For an arbitrary 2-valued σ-consistency problem a systematic search through a sorted argument power set would only need to store the current node in order to know which node to examine next. In which case, each node in the search would be of length n^2. Hence $O(n^2)$ is an upper bound for the space complexity for solving any 2-valued σ-consistency problem.

4. Complexity of 3-valued labellings

In order to formalise the 3-valued σ-consistency problem, we first directly translate the argument-set-based form of Def. 1 to accommodate partial data sets, adding the OFF label to represent those arguments in the wider data set that are unlabelled in a particular reinstatement labelling. The OFF label is essential since it allows a defeat relation R to be learned that is consistent with partial labellings. Def. 5 combines the reinstatement approach adopted in [6] alongside the concept of a signature $\Sigma_\sigma((A,R))$ from Def. 2.

Definition 5. (3-valued form) *Let A be a set of arguments; let $L : A \to \{\text{IN}, \text{UNDEC}, \text{OUT}, \text{OFF}\}$ be a labelling; let $R \subseteq A \times A$ be a defeat relation. $L \in \sigma((S,R))$, where $S = A \setminus X$, and where X is the set of arguments labelled* OFF, *iff each of the following conditions hold $\forall b \in S$:*

1. $L(b) = \text{IN} \iff \forall a \in S : (L(a) \neq \text{OUT} \implies (a,b) \notin R)$;
2. $L(b) = \text{UNDEC} \iff \forall a \in S : (L(a) = \text{IN} \implies (a,b) \notin R) \wedge \exists c \in A : (L(c) = \text{UNDEC} \wedge (c,b) \in R)$; *and*
3. $L(b) = \text{OUT} \iff \exists a \in S : (L(a) = \text{IN} \wedge (a,b) \in R)$.

Def. 6 formally presents the 3-valued σ-consistency problem of finding a solution defeat relation set consistent with input 3-valued argument acceptability data.

Definition 6. (3-valued σ-consistency problem) *Given a set of arguments A, a semantics σ, and a set of labellings T such that for each $L \in T$, $L : A \to \{\text{IN}, \text{UNDEC}, \text{OUT}, \text{OFF}\}$, is there a defeat relation $R \subseteq A \times A$ consistent with T such that $\forall L : (L \in \sigma((S,R)) \in \Sigma_\sigma((S,R)))$, where $S = A \setminus X$, and where X is the set of arguments labelled* OFF*?*

We again note that no assumption of a single solution defeat relation can be made in general (e.g. a labelling with two arguments labelled IN and one argument labelled OUT could be satisfied by multiple distinct defeat relations) and we gain insight into the complexity of the 3-valued problem by first observing the symmetry of inconsistent defeats in Theorem 3. It is important to note that, as was the case when we considered the ramifications of Theorem 1, Theorem 3 is not limited to symmetric AFs but also applies to asymmetric AFs. Theorem 3 outlines that for complete semantics if an arbitrary defeat $(a,b) \in R$ is incompatible with a labelling set, then $(b,a) \in R$ will also be incompatible.

Theorem 3. (Bidirectional defeat inconsistency for 3-valued labellings) *Assuming complete argumentation semantics, for any set of labellings $T : \forall L \in T$, $L : A \to \{\text{IN}, \text{OUT}, \text{UNDEC}, \text{OFF}\}$, we have $\exists a, b \in A : ((a,b) \in R \implies L \notin \sigma((S,R)) \iff (b,a) \in R \implies L \notin \sigma((S,R)))$, where $S = A \setminus X$, and X is the set of arguments labelled* OFF.

Proof. Suppose we have σ as complete semantics, a and b are arbitrary arguments in A, and $S = A \setminus X$, and where X is the set of arguments labelled OFF.
(*Forward implication*) Suppose $(a,b) \in R \implies L \notin \sigma((S,R))$. From Def. 5 we draw expressions of the form $L(a) \wedge L(b) \implies (a,b) \notin R$. We have three cases:
Case 1. $L(b) = \text{IN} \wedge L(a) = \text{IN} \implies (a,b) \notin R$. Suppose $(b,a) \in R$, then by the first condition of Def. 5 $L(a) = \text{IN} \implies (b,a) \notin R$. Contradiction!
Case 2. $L(b) = \text{IN} \wedge L(a) = \text{UNDEC} \implies (a,b) \notin R$. Suppose $(b,a) \in R$, then by the second condition of Def. 5 $L(a) = \text{UNDEC} \implies (b,a) \notin R$. Contradiction!
Case 3. $L(b) = \text{UNDEC} \wedge L(a) = \text{IN} \implies (a,b) \notin R$. Suppose $(b,a) \in R$, then by the first condition of Def. 5 $L(a) = \text{IN} \implies (b,a) \notin R$. Contradiction!
Therefore if $(a,b) \in R \implies L \notin \sigma((S,R))$ then $(b,a) \in R \implies L \notin \sigma((S,R))$.
(*Backward implication*) Since a and b are arbitrary, the backward implication immediately follows from the permutation $a \leftrightarrow b$ in the forward implication.
Therefore if $(b,a) \in R \implies L \notin \sigma((S,R))$ then $(a,b) \in R \implies L \notin \sigma((S,R))$. □

We use Algorithm 1 to indicate an upper bound for complexity in solving the 3-valued σ-consistency problem. The algorithm begins with a full initial defeat relation such that all arguments defeat all other arguments and then prunes the defeats that are incompatible with the acceptability data. The algorithm involves two passes through the data: one to remove defeats inconsistent with the labellings, and a second to check the resulting defeat relation R is consistent with the labellings.

Algorithm 1 A pruning algorithm that returns a defeat relation R that is consistent with a set of labellings T and argument set A or indicates that no such R is possible.

```
procedure DEFEAT PRUNING(A, T)
    R ← {(a,b), ∀a,b ∈ A}
    for ∀L ∈ T do
        for ∀a,b ∈ A do
            if (L(b) = IN ∧ L(a) ∈ {IN, UNDEC}) ∨
                  (L(b) = UNDEC ∧ L(a) = IN) then
                R ← R \ (a,b)
    for ∀L ∈ T do
        for ∀b ∈ A do
            if L(b) = UNDEC then
                if ∄a ∈ A : ((a,b) ∈ R) ∧ (L(a) = UNDEC) then
                    return failure
            if L(b) = OUT then
                if ∄a ∈ A : ((a,b) ∈ R) ∧ (L(a) = IN) then
                    return failure
    return R
```

We stress that Algorithm 1 is not the only method, and we certainly do not suggest it is optimal in terms of any performance metric, of addressing the 3-valued σ-consistency problem but, as Theorem 4 indicates, it shows the problem is significantly less complex to solve than its 2-valued peer. This result may appear unintuitive given the bijective mapping of 2-valued and 3-valued labellings in forward argumentation. The simple reason for the divergence in complexity between the two σ-consistency problems is due to the ambiguity of the NOTIN label which can be either OUT or UNDEC in the 3-valued approach. Theorem 3 shows that those defeats that are inconsistent with 3-valued data are direct consequences of the labels themselves, whereas in Theorem 1 we see that the labels of 2-valued data are insufficient to determine inconsistency and the existing defeat relation R must be examined also. This self-referential search process in finding a solution defeat relation R for the 2-valued σ-consistency problem is the cause of the additional complexity.

Theorem 4. (Defeat pruning for 3-valued σ-consistency problem) *The 3-valued σ-consistency problem can be solved by a defeat pruning algorithm with a time complexity of $O(n^2|T|)$ and a space complexity of $O(n^2)$ where n is the number of arguments and $|T|$ is the number of labellings.*

Proof. Let us define some R' that is the defeat relation output by Algorithm 1 upon receiving input T. There exist two cases for solving the 3-valued σ-consistency problem: (*Case 1 Forward implication*) Suppose there $\exists R$ that is consistent with T. Suppose R' is not consistent with T, then $\exists L \in T : (L \notin \sigma((S,R)))$. By the definition of R' it cannot be

that some inconsistent defeat is in R' and so there must be some essential defeat missing from R'. From Def. 5 it must be that $(\exists b_1 \in S : (L(b_1) = \text{UNDEC}) \wedge \forall c \in S : (L(c) = \text{UNDEC} \implies (c, b_1) \notin R')) \vee (\exists b_2 \in S : (L(b_2) = \text{OUT}) \wedge \forall a \in S : (L(a) = \text{IN} \implies (a, b_2) \notin R'))$. But since R exists then for any such $L(b_1) = \text{UNDEC}$ there is some appropriate $(c, b_1) \in R \setminus R'$, or for any $L(b_2) = \text{OUT}$ there is some appropriate $(a, b_2) \in R \setminus R'$. But by the definition of R' it must be that $R' \supseteq R$, contradiction! Therefore if there exists some R that is consistent with T then R' is also consistent with T.
(*Case 2 Backward implication*) Suppose there $\nexists R$ consistent with T. Then clearly R' is not consistent with T. Hence if $\nexists R$ consistent with T then R' is also not consistent with T. Therefore there $\exists R$ that is consistent with T iff R' is consistent with T.
To find R' it is necessary in the worst case to check all n^2 possible defeats for each $L \in T$. From Theorem 3 it is clear that for any defeat (a, b) to be evaluated for consistency with some $L \in T$ it is sufficient to simply check $L(a)$ and $L(b)$. Hence there are required at most $2n^2|T|$ operations required to produce R'. Similarly, once R' has been derived, there will be at most $2n^2|T|$ operations required to check that R' is consistent with T. Therefore the defeat pruning algorithm will find a solution defeat relation R' or prove that none exist in time complexity of $O(2n^2|T| + 2n^2|T|) = O(n^2|T|)$.
Finally, for each step in the process we need to store the current defeat set R, of size n^2. It follows that the space complexity is within $O(n^2)$. $\square$

5. σ-consistency For Other Semantics

Throughout this paper the focus has been on complete argumentation semantics. However, turning our attention to the alternative semantics as originally presented in [10] allows us to informally outline some relevant results.

For σ-consistency under stable semantics, we can quickly identify that both the 2-valued and 3-valued problems reduce to a special case of the 3-valued problem where no labelling contains UNDEC labelled arguments. Explicitly for the 2-valued problem this means that all NOTIN labelled arguments are interpreted as labelled OUT. Therefore the problem is solvable in $O(n^2|T|)$ time and $O(n^2)$ space complexity.

For σ-consistency under preferred semantics, we conjecture that the problem is not in **NP** unless **coNP** = **P**. The verification of any solution defeat relation is achieved by verifying each labelling in the data set. As outlined in [9,12], the verification of any labelling under preferred semantics is **coNP**-complete. We observe the **coNP**-complete result is derived from the special instance of verifying the empty set is a preferred extension, hence the result pertains to both the 2-valued and 3-valued variants. Further, as discussed in [13], verification results for all the major semantics hold for both 2-valued and 3-valued data. This means that solving σ-consistency is likely to be a hard problem.

For σ-consistency under grounded semantics, it is easy to see that for both 2-valued and 3-valued acceptability data, the problem is within **NP**, since the verification of each labelling in $|T|$ is within **P** [12]. However, we strongly conjecture that both 2-valued and 3-valued σ-consistency problems under grounded semantics are in fact **NP**-complete. There is not room to demonstrate a full proof in the confines of this paper. However, a proof similar to that used for Theorem 2 can be constructed by reducing from the Monotone 3SAT problem such that the two types of clauses are reduced by r thus:
Case 1: $c_i = (a_i \vee b_i \vee c_i)$ and $r(c_i) = (S, L_{1i}, L_{2i})$, where $S = \{a_i, b_i, c_i, \omega\}$, $L_{1i} = \{L(\omega) = \text{IN}\}$, and for 2-valued (resp. 3-valued) σ-consistency we have $L_{2i} = \{L(a_i) =$

NOTIN, $L(b_i) =$ NOTIN, $L(c_i) =$ NOTIN, $L(\omega) =$ NOTIN$\}$ (resp. $L_{2i} = \{L(a_i) =$ UNDEC, $L(b_i) =$ UNDEC, $L(c_i) =$ UNDEC, $L(\omega) =$ UNDEC$\}$);
Case 2: $c_i = (\neg a_i \vee \neg b_i \vee \neg c_i)$ and $r(c_i) = (S, L_{1i})$, where $S = \{a_i, b_i, c_i, \omega\}$, and for 2-valued (resp. 3-valued) σ-consistency we have $L_{1i} = \{L(a_i) =$ NOTIN, $L(b_i) =$ NOTIN, $L(c_i) =$ NOTIN, $L(\omega) =$ IN$\}$ (resp. $L_{2i} = \{L(a_i) =$ OUT, $L(b_i) =$ OUT, $L(c_i) =$ OUT, $L(\omega) =$ IN$\}$).

6. Related Work

Research on the topic of extension enforcement [2,3], is concerned with determining what additions could be made to a defeat relation in order to accommodate new extensions. However, there are notable departures from the direction pursued in this paper, such as requiring monotonic growth of the defeat relation, whereas we also allow reduction when solving for σ-consistency.

Argumentation realizability [4,11,17,20], extends beyond extension enforcement by removing the requirement of monotonic enlargement of the defeat relation R. Realizability requires that there exists a defeat relation that can express precisely the given set of interpretations (labellings or extensions), with no other interpretations expressible from the defeat relation. This assumption of completeness of the input extension/labelling set can be understood as a special case of σ-consistency where partial labellings pertaining to argument subgraphs are not permitted and the input labellings are exactly $\sigma((A,R))$. Interestingly, research into argumentation realizability has thus far encountered difficulty in determining a complexity class for complete semantics, remaining apparently unsolved despite its importance as a foundation for other semantics as previously discussed. Note that the complexity for realizability under complete semantics is conjectured to be **NP**-hard due to the association with MaxSat algorithms in deriving a solution.

Argumentation synthesis [18,19] develops the concept of realizability by relaxing the requirement for one-to-one mapping; the solution defeat set must satisfy a maximal number of argument labels. The approach further differentiates itself from realizability by accepting positive labels that are to be satisfied, but also negative labels that should not be satisfied. Argumentation synthesis is posed as an optimization problem that can accommodate noisy data sets, which will be in accordance with a wide set of real problems. Thus far in the literature, argumentation synthesis has only been applied to 2-valued problems (i.e. extension-based) and ignored partial labellings, unlike the general σ-consistency problems considered in this paper. As an optimization problem it requires its own complexity analysis appropriate for a Max-Sat search. Similar to realizability, complexity results for complete semantics have thus far been elusive albeit conjectured to belong to the **NP**-hard class of problems [19].

An alternative to argumentation synthesis for handling noisy data are the probability-based approaches [15,16,22] that do not overtly seek out minimising the number of misclassified errors as the singular goal. Whilst [15,16] use Bayesian inference and [22] uses the more elementary Kolmogorov's axioms, both focus on 2-valued argument acceptability data. It is notable that [15,16] suffer the problem of exponential complexity when determining their Bayesian calculations, since power sets of extension argument acceptabilities must be considered with a resulting combinatorial explosion. In contrast, [22] does not suffer from this same problem but has an altogether different dilemma in identifying from where the prior probabilities that are assigned to the argument rules are obtained, before these are mapped to the corresponding graph.

The most closely related research [21,23] examines the σ-consistency problem from the 2-valued and 3-valued perspectives but under grounded semantics. The complexity results from [21,23] claim that processing the 2-valued (resp. 3-valued) σ-consistency problem under grounded semantics is solvable in $O(n^2|T|)$ (resp. $O(n^3|T|)$) time (by our notation). These findings clearly disagree with our conjecture from Section 5 that both problems are **NP**-complete. We believe that the complexity results from [21,23] are incorrect and the error stems from neglecting the subset minimality of grounded semantics and the potential for the empty set to be the grounded labelling. A formal proof is forthcoming.

7. Concluding Remarks

We examined the computational complexity of the σ-consistency problem (where 'σ-consistency' refers to 'σ-consistency under complete semantics' throughout the paper) that determines whether a solution defeat relation exists that is wholly consistent with a set of argument acceptability labellings under the given semantics. The paper offers the following contributions.

- Introduced and defined the σ-consistency problem, which asks if a solution defeat relation exists that is consistent with a set of argument acceptability labellings.
- The 2-valued σ-consistency problem is proved to be **NP**-complete, and shown to have a space complexity within $O(n^2)$.
- The 3-valued σ-consistency problem is proved to be solvable by algorithms with a time complexity within $O(n^2|T|)$ and space complexity within $O(n^2)$.
- Provided informal complexity results for stable, preferred, and grounded semantics.
- The complexity results provide insight into the scaleability of algorithms designed to determine a defeat relation from argument acceptability data, suggesting a preference for sourcing 3-valued data to reduce the risk of solving high complexity problems.

Future work expanding the formal attention to other semantics, such as preferred or semi-stable, as well as to more advanced forms of argumentation, such as weighted or bipolar semantics, would also require rigorous complexity analysis of these forms in order to locate the expectations for relevant algorithms. We would also identify, as a fertile ground for exploration, the pursuit of theory underpinning the enumeration and/or counting of solution defeat relations under σ-consistency, as well as related research into the "quality" of solution defeat relations compared with the notion of a ground truth.

Acknowledgements

This work was partially supported by a PhD studentship from King's College London.

References

[1] V. Aleven. *Teaching case-based argumentation through a model and examples*. PhD thesis, University of Pittsburgh, 1997.

[2] Ringo Baumann. What does it take to enforce an argument? minimal change in abstract argumentation. In *ECAI*, volume 12, pages 127–132, 2012.

[3] Ringo Baumann and Gerhard Brewka. Expanding argumentation frameworks: Enforcing and monotonicity results. *COMMA*, 10:75–86, 2010.

[4] Ringo Baumann, Wolfgang Dvorák, Thomas Linsbichler, Hannes Strass, and Stefan Woltran. Compact argumentation frameworks. In *ECAI*, pages 69–74, 2014.

[5] Trevor JM Bench-Capon and Paul E Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10-15):619–641, 2007.

[6] Martin Caminada. On the issue of reinstatement in argumentation. In *European Workshop on Logics in Artificial Intelligence*, pages 111–123, 2006.

[7] Martin WA Caminada and Dov M Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109, 2009.

[8] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Symmetric argumentation frameworks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 317–328. Springer, 2005.

[9] Yannis Dimopoulos and Alberto Torres. Graph theoretical structures in logic programs and default theories. *Theoretical Computer Science*, 170(1-2):209–244, 1996.

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

[11] Paul E Dunne, Wolfgang Dvořák, Thomas Linsbichler, and Stefan Woltran. Characteristics of multiple viewpoints in abstract argumentation. *Artificial Intelligence*, 228:153–178, 2015.

[12] Paul E Dunne and Michael Wooldridge. Complexity of abstract argumentation. In *Argumentation in artificial intelligence*, pages 85–104. Springer, 2009.

[13] Wolfgang Dvorák and Paul E Dunne. Computational problems in formal argumentation and their complexity. *Handbook of formal argumentation*, 4, 2018.

[14] Michael R Garey and David S Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[15] Hiroyuki Kido and Beishui Liao. A Bayesian approach to direct and inverse abstract argumentation problems. *arXiv preprint arXiv:1909.04319*, 2019.

[16] Hiroyuki Kido and Frank Zenker. Argument-based Bayesian estimation of attack graphs: A preliminary empirical analysis. In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 523–532. Springer, 2017.

[17] T Linsbichler, J Puehrer, and H Strass. Characterizing realizability in abstract argumentation. In *Proceedings of the 16th International Workshop on Non-monotonic reasoning*, 2016.

[18] Andreas Niskanen, Daniel Neugebauer, Matti Järvisalo, and Jörg Rothe. Deciding acceptance in incomplete argumentation frameworks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2942–2949, 2020.

[19] Andreas Niskanen, Johannes Wallner, and Matti Järvisalo. Synthesizing argumentation frameworks from examples. *Journal of Artificial Intelligence Research*, 66:503–554, 2019.

[20] Jörg Pührer. Realizability of three-valued semantics for abstract dialectical frameworks. *Artificial Intelligence*, 278:103198, 2020.

[21] Régis Riveret. On learning abstract argumentation graphs from bivalent statement labellings. In *28th International Conference on Tools with Artificial Intelligence*, pages 190–195. IEEE, 2016.

[22] Régis Riveret, Pietro Baroni, Yang Gao, Guido Governatori, Antonino Rotolo, and Giovanni Sartor. A labelling framework for probabilistic argumentation. *Annals of Mathematics and Artificial Intelligence*, pages 1–51, 2018.

[23] Régis Riveret and Guido Governatori. On learning attacks in probabilistic abstract argumentation. In *AAMAS*, pages 653–661, 2016.

[24] Yining Wu, Martin Caminada, and Mikotaj Podlaszewski. A labelling-based justification status of arguments. *Studies in Logic*, 3(4):12–29, 2010.