



HAL
open science

Incomplete Bipolar Argumentation Frameworks

Marie-Christine Lagasquie-Schiex, Jean-Guy Mailly, Antonio Yuste-Ginel

► **To cite this version:**

Marie-Christine Lagasquie-Schiex, Jean-Guy Mailly, Antonio Yuste-Ginel. Incomplete Bipolar Argumentation Frameworks. IRIT/RR-2023-01-FR, IRIT : Institut de Recherche en Informatique de Toulouse. 2023, pp.1-24. hal-04104907

HAL Id: hal-04104907

<https://ut3-toulouseinp.hal.science/hal-04104907v1>

Submitted on 24 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Incomplete Bipolar Argumentation Frameworks

M-Christine Lagasquie-Schiex

IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3

Jean-Guy Mailly

Université Paris Cité, LIPADE, F-75006 Paris, France

Antonio Yuste-Ginel

Complutense University of Madrid, Spain

Tech. Report

IRIT/RR- -2023- -01- -FR

May 2023

Abstract

Dung's Abstract Argumentation Framework has been generalized in various directions. We combine two of these generalizations: Bipolar Argumentation Frameworks (BAFs), where a relation representing supports between arguments is added, and Incomplete Argumentation Frameworks (IAFs), where the existence of arguments and attacks may be uncertain. We discuss how the notion of completion of IAFs can be adapted to take into account the nature of the support relation, providing a couple of alternative definitions. After that, we analyse the impact of choosing among these alternatives on the complexity of argument acceptability problems. Finally, we give a logical encoding of our new framework in the Dynamic Logic of Propositional Assignments.

Contents

1	Introduction	1
2	Background	2
2.1	Abstract Argumentation Frameworks without uncertainty	2
2.2	Incomplete Argumentation Frameworks	4
2.3	The Dynamic Logic of Propositional Assignments	5
3	Incomplete Bipolar Argumentation Framework	6
4	Complexity of Reasoning with IBAFs	8
5	Logical encoding of IBAFs	10
6	Conclusion	13
	References	14
A	Proofs	17
A.1	Proofs for Section 4	17
A.2	Proofs for Section 5	19

1 Introduction

Formal argumentation has become an essential approach to reasoning in Artificial Intelligence in recent decades, finding applications in many contexts (e.g., in multi-agent systems [10]). These studies have been deeply influenced by the Abstract Frameworks (AFs) introduced by Dung in [20], where nodes of a graph are used to represent arguments while the edges represent an attack relation among them. In this context, different semantics are used for selecting *extensions* from a given AF, *i.e.* sets of arguments considered jointly acceptable because they satisfy some intuitive requirements.

Since the seminal work of Dung, many extensions of the original model have flourished. Here, we focus on two families of such generalisations. First, *the addition of new kinds of interactions among arguments*: among others, *support relations* ([32, 9, 33]), *higher-order frameworks* [5, 11] (where attacks might target other attacks, not only arguments), or *collective interactions* (where the source of attacks might be a set of nodes, instead of a single one). Second, *the addition of uncertainty to the model*, which can be done either by the introduction of *weights and preferences over arguments and interactions* [1, 4], or by taking into account *uncertainty about the presence of the different elements* (both in a qualitative [31] and a probabilistic fashion [26]).

More specifically, we focus on the computational study of the crossroads between one approach of each family: the addition of support relations (BAFs), and the consideration of qualitative uncertainty about the involved elements (IAFs). Our main reason to use BAFs is that they can be a practical and more realistic tool for representing some situations (an example for decision-making in the medical domain can be found in [27]). We restrict our attention to two specific interpretations of BAFs: necessary BAFs [32] and deductive BAFs [9]. Although other proposals can be found in the literature (e.g., evidential BAFs [33]), we ground our choice on simplicity (w.r.t. other proposals), and similarity among the chosen ones (there is a known duality between necessary and deductive supports [13]). As for the reason to use IAFs, it is obvious that arguments and their interactions cannot always be considered certain. This is particularly clear in multi-agent scenarios when an agent is not sure of the argumentative background information of its interlocutors. Additionally, human beings are able to reason with uncertain data and if we want formal argumentative tools for assisting them, then these tools must take into account the mentioned uncertainty. Finally, it is very important to know the impact of this uncertainty from a computational point of view, particularly when one wants to produce an efficient tool. The following example illustrates the kind of situations that motivates our study:

Example 1 *The pension reform wanted by the government is the main topic of a heated discussion between people. The following arguments are exchanged:*

a_0 : *The pension reform is important and must be implemented.*

a_1 : *Indeed. Because the pension financing system is in deficit (a_1 supports a_0).*

a_3 : *This reform is the only way to avoid a reduction in the amount of pensions (a_3 supports a_0).*

a_4 : *It would be surprising if this reform were the only way to avoid this reduction (a_4 attacks a_3).*

a_5 : *Indeed, an increase in contributions would also prevent a reduction in the amount of pensions (a_5 supports a_4).*

a_6 : *This reform is too premature; there are other reforms in progress and we do not yet know their impact (a_6 attacks a_0).*

Clearly, uncertainty and incompleteness exist in this exchange. First, several politicians consider that the deficit of the pension system is not the real motivation of the government for reforming (so the support from a_1 to a_0 would be uncertain); secondly, the argument a_4 is clearly a non-convincing argument; and finally, perhaps the impact of the previous reforms on the new one (so the attack from a_6 to a_0) may have already been considered by the government. \square

This paper defines and studies *Incomplete Bipolar AFs* (IBAFs), which allow for modelling cases like the one we just showed. Our contribution w.r.t. this new formalism is threefold. On the conceptual side, we provide two alternative definitions of *completions* (the hypothetical removal of uncertainty typically used to reason about IAFs). We discuss how these definitions encode two different intuitions: whether support is to be considered “prior” (in the sense of more important) to uncertainty, or the other way around. On the complexity side, we draw a map of how hard argument acceptability problems are w.r.t. our new model, hence showing what is the impact of choosing among the different notions of completion. More precisely, we show that assessing the acceptability of arguments has the same complexity as in the case of (non-bipolar) IAFs when uncertainty is prior to support, while the other option induces a higher complexity. Finally, we provide a logical encoding of these problems in the Dynamic Logic of Propositional Assignments (DL-PA) [2], a well-behaved variant of propositional dynamic logic that has been proven useful to reason about argumentation in recent years.

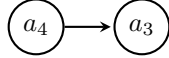
The rest of this paper is organized as follows: Sec. 2 gives the background on argumentation and on logics; the definition of IBAF is given in Sec. 3; the complexity results are presented in Sec. 4 and the logical encoding of IBAF in Sec. 5; Sec. 6 concludes the paper by giving some perspectives. Note that the proofs of our results can be found in Appendix A.

2 Background

2.1 Abstract Argumentation Frameworks without uncertainty

Basic notions of abstract argumentation. We suppose the existence of a finite set of arguments \mathbf{A} . An **argumentation framework** (AF) is a pair

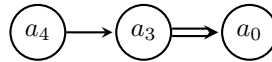
$\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ with $\mathcal{A} \subseteq \mathbf{A}$ the set of **arguments** and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ the set of **attacks**. For $a, b \in \mathcal{A}$, we say that a **attacks** b if $(a, b) \in \mathcal{R}$ (and we sometimes use the infix notation $a\mathcal{R}b$). If b attacks some $c \in \mathcal{A}$, then a **defends** c against b . Similarly, a set $S \subseteq \mathcal{A}$ attacks (resp. defends) an argument b if there is some $a \in S$ that attacks b (resp. if, for any $a\mathcal{R}b$, there is $c \in S$ that defends b against a). Let us consider Ex. 1 without taking into account the potential uncertainty, arguments a_4 and a_3 and their relationship can be represented by the graph:



We classically use the concept of **extensions** for evaluating the acceptability of arguments, *i.e.* sets of collectively acceptable arguments. The usual semantics are based on two main principles: conflict-freeness and admissibility. Given $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ an AF, the set $S \subseteq \mathcal{A}$ is **conflict-free** iff $\forall a, b \in S, (a, b) \notin \mathcal{R}$; S is **admissible** iff it is conflict-free and $\forall a \in S, \forall b \in \mathcal{A}$ s.t. $b\mathcal{R}a, \exists c \in S$ s.t. $c\mathcal{R}b$. We use $\text{cf}(\mathcal{F})$ (respectively $\text{ad}(\mathcal{F})$) to denote the set of conflict-free (resp. admissible) sets of an AF \mathcal{F} . We focus on the four semantics proposed by Dung. Formally, the admissible set $S \subseteq \mathcal{A}$ is: a **complete** extension iff S contains all the arguments that it defends; a **preferred** extension iff S is a \subseteq -maximal admissible set; a **grounded** extension iff S is a \subseteq -minimal complete extension; and a **stable extension** iff $S \in \text{cf}(\mathcal{F})$ and $\forall a \in \mathcal{A} \setminus S, S$ attacks a . We use $\text{co}(\mathcal{F})$, $\text{pr}(\mathcal{F})$, $\text{gr}(\mathcal{F})$ and $\text{st}(\mathcal{F})$ for the sets of (resp.) complete, preferred, grounded and stable extensions of \mathcal{F} (see more details in [20, 3]).

Bipolar argumentation framework. This notion has been initially defined as a general approach taking into account two kinds of interactions between arguments, a negative one (attacks) and a positive one (supports), see [12].

A **Bipolar Argumentation Framework** (BAF) is a tuple $\mathcal{B} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ where $\mathcal{A} \subseteq \mathbf{A}$ are arguments, $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation, and $\mathcal{S} \subseteq \mathcal{A} \times \mathcal{A}$ is a **support relation** (when $a\mathcal{S}b$ we say that a *supports* b). Given a (support) relation \mathcal{S} , we use \mathcal{S}^+ to denote its transitive closure (*i.e.*, the smallest (w.r.t. \subseteq) transitive relation containing \mathcal{S}). Let us consider Ex. 1, arguments a_4 , a_3 and a_0 and their relationship (ignoring uncertainty) can be represented by the graph:



In the general approach to BAFs, semantics are defined using the addition of new attacks. Nevertheless, it turned out that such a general approach is not sufficient for encoding some real cases and sometimes the drawback is the lack of guidelines for choosing the appropriate definitions and semantics depending on the application. Consequently, various kinds of support relations have been defined in the literature as specializations of this general framework. Among others, one could mention the notion of necessary support [32], deductive support [9], evidential support [33], backing support [14], and monotonic support

[24]. Here, we just focus on the two former notions, which have the following intuitive meaning: if a necessarily (resp. deductively) supports b then the acceptance of a is necessary for (resp. implies) the acceptance of b . Moreover, a duality exists between these two approaches: a necessarily supports b iff b deductively supports a (see [13]); so a deductive BAF is a necessary BAF in which the direction of the support arrows has been reversed (and vice-versa).

When the type of support is chosen, the reasoning is made once again with the notion of extension via the addition of new attacks. We focus on the necessary interpretation, as the deductive one follows from the mentioned duality by simply reversing support arrows. Let $\mathcal{B} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ be a BAF, let $a, b \in \mathcal{A}$, a **attacks b according to the necessary interpretation** iff either $a\mathcal{R}b$ (*Case 0: an existing direct attack*), or there is $c \in \mathcal{A}$ s.t. $a\mathcal{R}c$ and $c\mathcal{S}^+b$ (*Case 1: a new attack*), or there is $c \in \mathcal{A}$ s.t. $c\mathcal{R}b$ and $c\mathcal{S}^+a$ (*Case 2: a new attack*). The following figure illustrates cases 1 and 2:



Let *nec* and *ded* stand for ‘necessary’ and ‘deductive’, and let $t \in \{\text{nec}, \text{ded}\}$, we denote by \mathcal{R}^{nec} (resp. \mathcal{R}^{ded}) the set of attacks according to the necessary (resp. deductive) interpretation. Given $\mathcal{B} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ a BAF, $\sigma^t(\mathcal{B}) = \sigma(\langle \mathcal{A}, \mathcal{R}^t \rangle)$ **is the set of extensions of the BAF under the interpretation t** . We say that $a \in \mathcal{A}$ is **credulously accepted** w.r.t. σ and t if it belongs to some extension in $\sigma^t(\mathcal{B})$, and **skeptically accepted** if it belongs to each extension. Obviously, the new attacks are added in \mathcal{R} and can therefore be used in turn to create new other attacks through a saturation process.

From a computational point of view, BAFs under the necessary and deductive interpretations of support have the same complexity as standard AFs (see [16, 28, 21]).

2.2 Incomplete Argumentation Frameworks

Incomplete Argumentation Frameworks [15, 7, 6, 31] are AFs with qualitative uncertainty about the presence of some arguments or attacks. Formally, an **Incomplete Argumentation Framework** (IAF) is a tuple $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$ where: $\mathcal{A} \subseteq \mathbf{A}$ is the set of **certain arguments**; $\mathcal{A}^? \subseteq \mathbf{A}$ is the set of **uncertain arguments**; $\mathcal{R} \subseteq (\mathcal{A} \cup \mathcal{A}^?) \times (\mathcal{A} \cup \mathcal{A}^?)$ the set of **certain attacks**; and $\mathcal{R}^? \subseteq (\mathcal{A} \cup \mathcal{A}^?) \times (\mathcal{A} \cup \mathcal{A}^?)$ the set of **uncertain attacks**. \mathcal{A} and $\mathcal{A}^?$ are disjoint sets of arguments, and $\mathcal{R}, \mathcal{R}^?$ are disjoint sets of attacks. Intuitively, \mathcal{A} and \mathcal{R} correspond, respectively, to arguments and attacks that certainly exist, while $\mathcal{A}^?$ and $\mathcal{R}^?$ are those that may (or may not) actually exist.

Reasoning with such IAFs is generally made through the notion of completion, *i.e.* a classical AF that represents a “possible world” with respect to the uncertain information encoded in the IAF. Formally, given $\mathcal{I} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^? \rangle$ an IAF, a **completion** of \mathcal{I} is an AF $\langle \mathcal{A}_c, \mathcal{R}_c \rangle$ such that $\mathcal{A} \subseteq \mathcal{A}_c \subseteq \mathcal{A} \cup \mathcal{A}^?$ and $\mathcal{R} \cap (\mathcal{A}_c \times \mathcal{A}_c) \subseteq \mathcal{R}_c \subseteq (\mathcal{R} \cup \mathcal{R}^?) \cap (\mathcal{A}_c \times \mathcal{A}_c)$.

Finally, reasoning tasks like credulous acceptance, skeptical acceptance or verification are defined with respect to some or each completion [7, 6]: indeed

each classical reasoning task has two variants, following the possible view (the property holds in some completion) and the necessary view¹ (the property holds in each completion). These reasoning tasks are, in most cases, computationally harder than their counterpart for standard AFs (under the usual assumption that the polynomial hierarchy does not collapse) [7, 6]. This can be explained by the exponential number of completions. For instance, the acceptance problem for the grounded semantics is NP-c (or coNP-c) in the case of IAFs whereas it is P-c for simple AFs (see [6, 31] for more details).

2.3 The Dynamic Logic of Propositional Assignments

Syntax. We assume the existence of a denumerable set of propositional variables $\text{Prp} = \{p_1, p_2, \dots\}$. We suppose that Prp contains several kinds of distinguished variables capturing the statuses of arguments and relations between them. First, given a set of arguments $\mathcal{A} \subseteq \mathbf{A}$ we define its set of **awareness variables**, $\text{AW}_{\mathcal{A}} = \{\text{aw}_x \mid x \in \mathcal{A}\}$, and its set of **acceptance variables** $\text{IN}_{\mathcal{A}} = \{\text{in}_x \mid x \in \mathcal{A}\}$. Second, given a relation $X \subseteq \mathbf{A} \times \mathbf{A}$ we define its set of **attack variables** $\text{ATT}_X = \{\text{r}_{x,y} \mid (x,y) \in X\}$ and its set of **support variables** $\text{SUP}_X = \{\text{s}_{x,y} \mid (x,y) \in X\}$. Summing up, we assume that $\text{Prp}_{\mathbf{A}} \subseteq \text{Prp}$ where:

$$\text{Prp}_{\mathbf{A}} = \text{AW}_{\mathbf{A}} \cup \text{IN}_{\mathbf{A}} \cup \text{ATT}_{\mathbf{A} \times \mathbf{A}} \cup \text{SUP}_{\mathbf{A} \times \mathbf{A}}.$$

Formulas and programs of DL-PA are defined by mutual recursion:

$$\begin{aligned} \text{For formulas: } \varphi &::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid [\pi]\varphi, \\ \text{For programs: } \pi &::= +p \mid -p \mid \varphi? \mid (\pi; \pi) \mid (\pi \cup \pi) \mid \pi^*, \end{aligned}$$

where p ranges over Prp .

The intended meaning of formulas is as usual for atoms and the Boolean connectors. As for modal formulas, $[\pi]\varphi$ reads “ φ is true after every possible execution of π ”, so that the dual $\langle \pi \rangle \varphi$, defined as $\neg[\pi]\neg\varphi$, reads “there is a possible execution of π that makes φ true”. As for programs, their intended meaning is as follows: $+p$ (resp. $-p$) is the atomic program that makes p true (resp. false). $\varphi?$ is the program that tests whether φ is true. $(\pi; \pi')$ is the sequential composition of π and π' (“first execute π and then π' ”). $(\pi \cup \pi')$ is the non-deterministic choice (“choose non-deterministically between π or π' and execute one of them”). Finally, π^* is the unbounded iteration of π (“execute π a finite number of times”).

Semantics. Given a propositional valuation $v \subseteq \text{Prp}$ (so v is the set of the variables that are true), **truth for formulas** φ and the **meaning of programs** $\|\pi\|$ is given by mutual recursion:

$$\begin{aligned} v \models p & \quad \text{if } p \in v, \\ v \models [\pi]\varphi & \quad \text{if } (v, v') \in \|\pi\| \text{ implies } v' \models \varphi, \end{aligned}$$

¹We are aware that we use the word “necessary” with two different meanings. We choose not to deviate from the standard terminology in the literature. However it will be clear from the context if we mean “necessary support” or “necessary in all the completions”.

and as usual for the Boolean connectives; the interpretation of programs is:²

$$\begin{aligned} \|\!| +p \|\!| &= \{(v, v') \mid v' = v \cup \{p\}\}, & \|\!| -p \|\!| &= \{(v, v') \mid v' = v \setminus \{p\}\}, \\ \|\!| \varphi? \|\!| &= \{(v, v) \mid v \models \varphi\}, & \|\!| \pi; \pi' \|\!| &= \|\!| \pi \|\!| \circ \|\!| \pi' \|\!|, \\ \|\!| \pi \cup \pi' \|\!| &= \|\!| \pi \|\!| \cup \|\!| \pi' \|\!|, & \|\!| \pi^* \|\!| &= \|\!| \pi \|\!|^*. \end{aligned}$$

Here are some useful abbreviations in our object language (where $P = \{p_1, \dots, p_n\}$ is a finite subset of Prp):

$$\begin{aligned} \text{mkTrueSome}(P) &= \dot{\bigvee}_{p \in P} (+p \cup \text{skip}) = (+p_1 \cup \text{skip}); \dots; (+p_n \cup \text{skip}), \\ \text{mkFalseAll}(P) &= \dot{\bigvee}_{p \in P} (-p) = -p_1; \dots; -p_n, \\ \text{vary}(P) &= \dot{\bigvee}_{p \in P} (+p \cup -p) = (+p_1 \cup -p_1); \dots; (+p_n \cup -p_n), \\ \text{if } \varphi \text{ then } \pi \text{ else } \pi' &= (\varphi?; \pi) \cup (\neg\varphi?; \pi'), \\ \text{while } \varphi \text{ do } \pi &= (\varphi?; \pi)^*; \neg\varphi?. \end{aligned}$$

From valuations to (B)AFs and backward. From our hypothesis that Prp contains $\text{Prp}_{\mathbf{A}}$, we can define for each valuation v the BAF $\langle \mathcal{A}_v, \mathcal{R}_v, \mathcal{S}_v \rangle$ (called the **BAF represented by** v) where: $\mathcal{A}_v = \{x \in \mathbf{A} \mid \text{aw}_x \in v\}$, $\mathcal{R}_v = \{(x, y) \in \mathcal{A}_v \times \mathcal{A}_v \mid \mathbf{r}_{x,y} \in v\}$, and $\mathcal{S}_v = \{(x, y) \in \mathcal{A}_v \times \mathcal{A}_v \mid \mathbf{s}_{x,y} \in v\}$. The other way round, each BAF $\langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ is represented by **its associated valuation** $v_{\langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle} = \{\text{aw}_x \mid x \in \mathcal{A}\} \cup \{\mathbf{r}_{x,y} \mid (x, y) \in \mathcal{R}\} \cup \{\mathbf{s}_{x,y} \mid (x, y) \in \mathcal{S}\}$. Note that both functions (from valuations to BAFs and backward) can be also defined for AFs, by just ignoring the supports. Finally, for each valuation v we define the **extension associated to** v as the set $E_v = \{x \in \mathbf{A} \mid \text{in}_x \in v\}$.

Argumentation semantics in DL-PA. There are some papers approaching the encoding of Dung's AFs semantics and some of its extensions in DL-PA [17, 19, 18, 25, 35]. Regarding abstract argumentation semantics, here is the list of those that have been captured in DL-PA so far: admissible, stable, complete, preferred, grounded, semi-stable, eager, stage, and naive. The main idea underlying these encodings is to write a generic DL-PA program mkExt^σ parametrised by each semantics σ , s.t. for every AF $\langle \mathcal{A}, \mathcal{R} \rangle$ we have that: $\sigma(\langle \mathcal{A}, \mathcal{R} \rangle) = \{E_{v'} \mid (v_{\langle \mathcal{A}, \mathcal{R} \rangle}, v') \in \|\!| \text{mkExt}^\sigma \|\!|\}$. For our purposes, we rely on the previous encoding of stable, complete, preferred, and grounded semantics (see [35, Theorem 1]). However, and due to space reasons, we just include here the instance of mkExt for capturing stable semantics as an illustration:

$$\begin{aligned} \text{Well} &= \bigwedge_{x \in \mathbf{A}} (\text{in}_x \rightarrow \text{aw}_x), \\ \text{Stable} &= \text{Well} \wedge \bigwedge_{x \in \mathbf{A}} \left(\text{aw}_x \rightarrow (\text{in}_x \leftrightarrow \neg \bigvee_{y \in \mathbf{A}} (\text{in}_y \wedge \mathbf{r}_{y,x})) \right), \\ \text{mkExt}^{\text{st}} &= \text{vary}(\text{IN}_{\mathbf{A}}); \text{Stable}. \end{aligned}$$

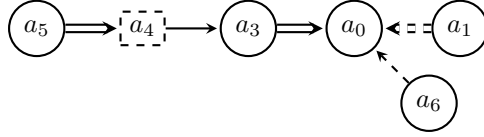
3 Incomplete Bipolar Argumentation Framework

Now we introduce our new *Incomplete Bipolar Argumentation Frameworks*, which generalizes both BAFs and IAFs.

²Given a binary relation R , we write R^* the reflexive and transitive closure of R .

Definiton 1 (Incomplete Bipolar AF) An Incomplete Bipolar Argumentation Framework (IBAF) is a tuple $\mathcal{IB} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^?, \mathcal{S}, \mathcal{S}^? \rangle$, where $\mathcal{A}, \mathcal{A}^?$ are disjoint sets of arguments and $\mathcal{R}, \mathcal{R}^?, \mathcal{S}, \mathcal{S}^?$ are disjoint relations between arguments.³

Example 2 The following figure gives a possible representation of the discussion described in Ex. 1 considering a deductive meaning for the support and the uncertainty of some elements (represented through dashed lines):



Before proposing a definition for the notion of completion, some points must be clarified. Let us consider the deductive⁴ support from a_5 to a_4 (i.e. if a_5 is accepted then a_4 must be accepted). A first approach can be to simply extend the notion of completion defined for IAF saying that an uncertain element can be present or not. We call this approach *plain* (or *pla*, for short). In this case, the acceptance of a_5 implies the acceptance of a_4 whenever a_4 is in the current completion; this is a *conditionally certain support*. A second possible approach is to consider that the meaning of support implies a constraint that should be satisfied in any completion if this support is certain: a completion in which a_5 would be accepted and not a_4 should be ruled out. Thus this second approach proposes to consider *unconditionally certain support*, giving, therefore, some kind of “priority” to the notion of support over uncertainty. The difficulty here is that the specificity we want to take into account is not only a syntactical one (an uncertain element is present or not) but also a semantic one (the meaning of the support must be preserved). This leads to proposing a more complex definition that brings out these different nuances:

Definiton 2 (IBAF Completions) Let $\mathcal{IB} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^?, \mathcal{S}, \mathcal{S}^? \rangle$ be an IBAF, let $\mathbf{t} \in \{\text{pla}, \text{nec}, \text{ded}\}$, and let σ be a semantics. A \mathbf{t} -completion of \mathcal{IB} w.r.t. σ is a BAF $\mathcal{B} = \langle \mathcal{A}_c, \mathcal{R}_c, \mathcal{S}_c \rangle$ such that:

1. $\mathcal{A} \subseteq \mathcal{A}_c \subseteq \mathcal{A} \cup \mathcal{A}^?$; $\mathcal{R} \cap (\mathcal{A}_c \times \mathcal{A}_c) \subseteq \mathcal{R}_c \subseteq (\mathcal{R} \cup \mathcal{R}^?) \cap (\mathcal{A}_c \times \mathcal{A}_c)$; $\mathcal{S} \cap (\mathcal{A}_c \times \mathcal{A}_c) \subseteq \mathcal{S}_c \subseteq (\mathcal{S} \cup \mathcal{S}^?) \cap (\mathcal{A}_c \times \mathcal{A}_c)$; and
2. if $\mathbf{t} \in \{\text{nec}, \text{ded}\}$ then $\forall (a, b) \in \mathcal{S}, \forall E \in \sigma^{\mathbf{t}}(\mathcal{B})$: (i) if $\mathbf{t} = \text{nec}$ and $b \in E$, then $a \in E$; and (ii) if $\mathbf{t} = \text{ded}$ and $a \in E$, then $b \in E$.

We denote as $\text{completions}_{\sigma}^{\mathbf{t}}(\mathcal{IB})$ the set of all completions of \mathcal{IB} under the \mathbf{t} interpretation and the semantics σ .

In Def. 2, the first condition represents the syntactical impact of the uncertainty (an uncertain element is present or not in the completion), whereas the

³Note that this constraint could be relaxed by permitting $\mathcal{R}^?$ and $\mathcal{S}^?$ to have a (possibly) non-empty intersection. It will be the subject of future work.

⁴From the mentioned duality, the same applies to necessary supports.

second condition specifies the semantic impact of the support on the building of the completion. Obviously, if $t = \text{pla}$, then the second condition is not applicable and so no constraint is added. Hence, the notation $\text{completions}_\sigma^{\text{pla}}(\mathcal{IB})$ can be simplified into $\text{completions}^{\text{pla}}(\mathcal{IB})$. We note by passing the strong relationship between the notion of deductive/necessary completions and the recently proposed *constrained IAFs* [29, 25], and *IAFs with dependencies* [22, 23]. However, the constraints or dependencies in these related works only take into account syntactical information, while we introduce semantic information as well.

Example 3 *Using Def. 2, the IBAF from Ex 1 has eight pla-completions, described at Figure 1. Moreover, with the preferred semantics, only 4 ded-completions can be built: $\mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_7$ and \mathcal{B}_8 :*

- for \mathcal{B}_3 (resp. \mathcal{B}_7), there is one preferred extension $\{a_0, a_1, a_4, a_5, a_6\}$; note that an additional attack from a_5 to a_3 is introduced for taking into account the deductive meaning of the support;
- for \mathcal{B}_4 , there is one preferred extension $\{a_1, a_4, a_5, a_6\}$; note that two additional attack are introduced: (a_5, a_3) and (a_6, a_3) ;
- for \mathcal{B}_8 , there is one preferred extension $\{a_4, a_5, a_6\}$; note that 3 additional attack exist: (a_5, a_3) , (a_6, a_3) and (a_6, a_1) .

For the other \mathcal{B}_i , a_5 belongs to the preferred extension whereas a_4 does not since a_4 is not in the completion, so they are not ded-completions.

Interestingly, the notion of “unconditional certainty” is useless when considering attacks. Indeed if an attack from a to b is certain whereas a or b are not, then only two kinds of completion exist: some completions contain a , b and the attack and so a and b cannot be accepted together; some others in which a or b are missing, so the attack too, and a and b cannot be accepted together; thus in each case, the meaning of the attack is satisfied with only a syntactical criterion.

4 Complexity of Reasoning with IBAFs

Let us investigate the complexity of reasoning with IBAFs. We focus on acceptability problems, *i.e.* possible credulous acceptability (PCA), necessary credulous acceptability (NCA) and their counterparts for skeptical acceptability (PSA and NSA). Formally, given an IBAF $\mathcal{IB} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^?, \mathcal{S}, \mathcal{S}^? \rangle$ and $a \in \mathcal{A}$, and given $t_1 \in \{\text{pla}, \text{nec}, \text{ded}\}$ and $t_2 \in \{\text{nec}, \text{ded}\}$ s.t. if $t_1 \in \{\text{nec}, \text{ded}\}$, then $t_1 = t_2$,

- $\sigma^{t_2-t_1}\text{-PCA}$: $\exists \mathcal{B} \in \text{completions}_\sigma^{t_1}(\mathcal{IB}), \exists E \in \sigma^{t_2}(\mathcal{B})$ s.t. $a \in E$?
- $\sigma^{t_2-t_1}\text{-NCA}$: $\forall \mathcal{B} \in \text{completions}_\sigma^{t_1}(\mathcal{IB}), \exists E \in \sigma^{t_2}(\mathcal{B})$ s.t. $a \in E$?
- $\sigma^{t_2-t_1}\text{-PSA}$: $\exists \mathcal{B} \in \text{completions}_\sigma^{t_1}(\mathcal{IB}), \forall E \in \sigma^{t_2}(\mathcal{B})$ s.t. $a \in E$?
- $\sigma^{t_2-t_1}\text{-NSA}$: $\forall \mathcal{B} \in \text{completions}_\sigma^{t_1}(\mathcal{IB}), \forall E \in \sigma^{t_2}(\mathcal{B})$ s.t. $a \in E$?

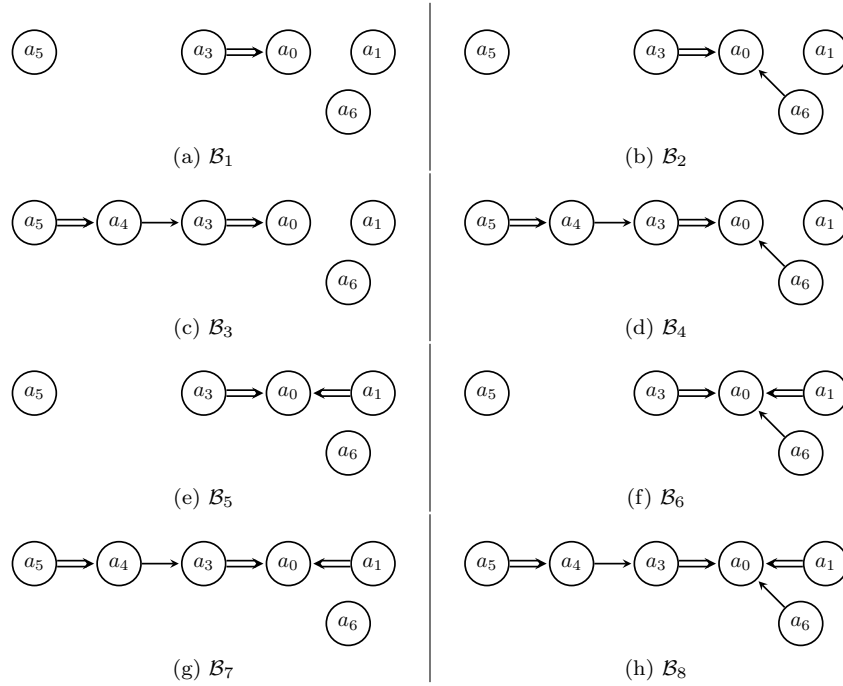


Figure 1: The completions of \mathcal{IB} from Example 1

In the first following propositions we consider the plain approach to computing completions (*i.e.* $\mathbf{t}_1 = \mathbf{pla}$), with the support relation being either the necessary or deductive support (*i.e.* $\mathbf{t}_2 \in \{\mathbf{nec}, \mathbf{ded}\}$). We show that checking the (various kinds of) acceptability of arguments is in this case not harder than in the case of (non-bipolar) IAFs. This comes from the following observation:

Observation 1 *Let $\mathcal{IB} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^?, \mathcal{S}, \mathcal{S}^? \rangle$ be an IBAF, and $\mathcal{B}^* = \langle \mathcal{A}^*, \mathcal{R}^*, \mathcal{S}^* \rangle$. Checking whether \mathcal{B}^* is a plain completion of \mathcal{IB} can be done in polynomial time (w.r.t. the number of arguments $|\mathcal{A} \cup \mathcal{A}^?|$).*

Proposition 1 *For $\mathbf{t}_2 \in \{\mathbf{nec}, \mathbf{ded}\}$ and $\sigma \in \{\mathbf{ad}, \mathbf{gr}, \mathbf{st}, \mathbf{co}, \mathbf{pr}\}$, $\sigma^{\mathbf{t}_2}$ -pla-PCA is NP-complete.*

Notice that \emptyset is always admissible in a BAF, so no argument is skeptically accepted, which means that the problem is trivial like in the case of IAFs.

Proposition 2 *For $\mathbf{t}_2 \in \{\mathbf{nec}, \mathbf{ded}\}$ and $\sigma \in \{\mathbf{gr}, \mathbf{st}, \mathbf{co}\}$, $\sigma^{\mathbf{t}_2}$ -pla-NSA is coNP-complete, it is trivial for $\sigma = \mathbf{ad}$, and it is Π_2^P -complete for $\sigma = \mathbf{pr}$.*

Proposition 3 *For $\mathbf{t}_2 \in \{\mathbf{nec}, \mathbf{ded}\}$ and $\sigma \in \{\mathbf{ad}, \mathbf{st}, \mathbf{co}, \mathbf{pr}\}$, $\sigma^{\mathbf{t}_2}$ -pla-NCA is Π_2^P -complete, and it is coNP-complete for $\sigma = \mathbf{gr}$.*

The last result concerns σ -PSA. Recall that it is trivial when $\sigma = \text{ad}$, as explained previously for ad-NSA .

Proposition 4 *For $t_2 \in \{\text{nec}, \text{ded}\}$ and $\sigma \in \{\text{co}, \text{gr}\}$, σ^{t_2} -pla-PSA is NP-complete, it is Σ_2^P -complete for $\sigma = \text{st}$, and Σ_3^P -complete for $\sigma = \text{pr}$.*

The situation is not so simple when $t_1 \in \{\text{nec}, \text{ded}\}$, since verifying whether a BAF is a t_1 -completion of an IBAF requires to look at the content of extensions, which is generally hard. The only exception is the grounded semantics.

Proposition 5 *Let $\mathcal{IB} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^?, \mathcal{S}, \mathcal{S}^? \rangle$ be an IBAF, and $\mathcal{B}^* = \langle \mathcal{A}^*, \mathcal{R}^*, \mathcal{S}^* \rangle$. Checking whether $\mathcal{B}^* \in \text{completions}_{\text{gr}}^t(\mathcal{IB})$ (for $t \in \{\text{nec}, \text{ded}\}$) can be done in polynomial time (w.r.t. the number of arguments $|\mathcal{A} \cup \mathcal{A}^?|$).*

Corollary 1 *For $t \in \{\text{nec}, \text{ded}\}$, gr^t -t-PCA is NP-complete, gr^t -t-NSA is coNP-complete, gr^t -t-NCA is coNP-complete and gr^t -t-PSA is NP-complete.*

We assume these problems to be harder for the other semantics. In particular, we can give some upper bounds for $\sigma \in \{\text{ad}, \text{co}, \text{st}\}$.

Proposition 6 *Let $\mathcal{IB} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^?, \mathcal{S}, \mathcal{S}^? \rangle$ be an IBAF, and $\mathcal{B}^* = \langle \mathcal{A}^*, \mathcal{R}^*, \mathcal{S}^* \rangle$. Checking whether $\mathcal{B}^* \in \text{completions}_{\sigma}^t(\mathcal{IB})$ (for $\sigma \in \{\text{ad}, \text{co}, \text{st}\}$ and $t \in \{\text{nec}, \text{ded}\}$) is a Δ_2^P problem.*

This induces that the acceptability problems are likely to be harder than their counterpart with $t_1 = \text{pla}$. However, proving that these are complete problems (*i.e.* that the provided upper bound is actually a tight complexity result) is still an open question, as well as the complexity of the various problems for $\sigma = \text{pr}$.

5 Logical encoding of IBAFs

Necessary and deductive supports in DL-PA. We start the original part of our encoding process by looking for DL-PA programs and formulas that capture the relevant notions of BAFs under the necessary and deductive interpretations of supports. In order to do so, we need to capture, following what we said in Sec. 2.1, the new kinds of attacks added under each of the interpretations. The first step is then to compute the transitive closure of the support relation in DL-PA, since this plays a crucial role in the definition of added attacks. The following program, adapted from [18], computes one step of such closure:

$$\text{step} = ;_{x,y,z \in \mathbf{A}} \left((\mathbf{aw}_x \wedge \mathbf{aw}_y \wedge \mathbf{aw}_z \wedge \mathbf{s}_{x,y} \wedge \mathbf{s}_{y,z})^? ; +\mathbf{s}_{x,z} \right).$$

Moreover, the following formula is true in those valuations where \mathcal{S}_v is transitive:

$$\text{Transitive} = \bigwedge_{x,y,z \in \mathbf{A}} \left((\mathbf{aw}_x \wedge \mathbf{aw}_y \wedge \mathbf{aw}_z \wedge \mathbf{s}_{x,y} \wedge \mathbf{s}_{y,z}) \rightarrow \mathbf{s}_{x,z} \right).$$

With both components, we can easily write a DL-PA program that computes the transitive closure of \mathcal{S}_v : $\text{transClosure} = \text{while } \neg \text{Transitive} \text{ do step}$.

Now, we are able to define DL-PA formulas that capture complex attacks (we also define a shorthand for capturing elements of \mathcal{R}_v):⁵

$$\begin{aligned} \mathbf{r}_{x,y}^{\text{aw}} &= \text{aw}_x \wedge \text{aw}_y \wedge \mathbf{r}_{x,y}, \\ \mathbf{r}_{x,y}^{\text{case1}} &= \text{aw}_x \wedge \text{aw}_y \wedge \bigvee_{z \in \mathbf{A}} (\text{aw}_z \wedge \mathbf{r}_{x,z} \wedge [\text{transClosure}]s_{z,y}), \\ \mathbf{r}_{x,y}^{\text{case2}} &= \text{aw}_x \wedge \text{aw}_y \wedge \bigvee_{z \in \mathbf{A}} (\text{aw}_z \wedge \mathbf{r}_{z,y} \wedge [\text{transClosure}]s_{z,x}). \end{aligned}$$

With these complex attacks already present in our syntactic repertoire, we can find a program that adds them to the BAF associated to v :

$$\begin{aligned} \text{addAttacks}^{\text{nec}} = & \ ;_{x,y \in \mathbf{A}} (\text{if } (\mathbf{r}_{x,y}^{\text{aw}} \vee \mathbf{r}_{x,y}^{\text{case1}} \vee \mathbf{r}_{x,y}^{\text{case2}}) \text{ then } +\mathbf{r}'_{x,y}; \text{ else skip}); \\ & \text{mkFalseAll}(\text{ATT}_{\mathbf{A} \times \mathbf{A}}); \\ & \ ;_{x,y \in \mathbf{A}} (\text{if } \mathbf{r}'_{x,y} \text{ then } +\mathbf{r}_{x,y} \text{ else skip}). \end{aligned}$$

Note that this program makes use of a set of fresh copies of attack variables $\text{ATT}'_{\mathbf{A} \times \mathbf{A}} = \{\mathbf{r}'_{x,y} \mid (x,y) \in \mathbf{A} \times \mathbf{A}\}$, whose existence can be assumed because Prp is countably infinite while \mathbf{A} is finite.

Using the duality between deductive and necessary support that we mentioned in Sec. 2.1, we can easily go from one interpretation to the other by means of the following DL-PA program (just as before, we make use of fresh copies of variables, this time with supports):

$$\begin{aligned} \text{deductive2necessary} = & \ ;_{x,y \in \mathbf{A}} (\text{if } s_{y,x} \text{ then } +s'_{x,y}; \text{ else skip}); \\ & \text{mkFalseAll}(\text{SUP}_{\mathbf{A} \times \mathbf{A}}); \\ & \ ;_{x,y \in \mathbf{A}} (\text{if } s'_{x,y} \text{ then } +s_{x,y}; \text{ else skip}). \end{aligned}$$

Hence, we can abbreviate $\text{addAttacks}^{\text{ded}} = \text{deductive2necessary}; \text{addAttacks}^{\text{nec}}$. We state formally the correctness of our encoding:

Proposition 7 *Let $\mathcal{B} = \langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ be a BAF, let $t \in \{\text{nec}, \text{ded}\}$, and let σ be an argumentation semantics, we have that*

$$\sigma^t(\mathcal{B}) = \{E_{v'} \mid (v_{\mathcal{B}}, v') \in \|\text{addAttacks}^t; \text{mkExt}^\sigma\|\}.$$

Computing completions of IBAFs in DL-PA. Moving to IBAFs, we first want to design a DL-PA program, $\text{mkComp}^{\text{pla}}(\mathcal{IB})$, that computes all the plain completions of a given $\mathcal{IB} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^?, \mathcal{S}, \mathcal{S}^? \rangle$. For doing so, we first define the **valuation associated to \mathcal{IB} , which is determined by its certain part**: $v_{\mathcal{IB}} = v_{\langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle}$. Note that $\langle \mathcal{A}_{v_{\mathcal{IB}}}, \mathcal{R}_{v_{\mathcal{IB}}}, \mathcal{S}_{v_{\mathcal{IB}}} \rangle$ is already a completion of \mathcal{IB} : it is the smallest one, where only certain arguments and certain attacks and supports between them are considered. In order to compute all the completions of \mathcal{IB} we make true subsets of propositional variables representing arguments in $\mathcal{A}^?$, attacks in $\mathcal{R}^?$, and supports in $\mathcal{S}^?$:

$$\text{mkComp}^{\text{pla}}(\mathcal{IB}) = \text{mkTrueSome}(\text{AW}_{\mathcal{A}^?}); \text{mkTrueSome}(\text{ATT}_{\mathcal{R}^?}); \text{mkTrueSome}(\text{ATT}_{\mathcal{S}^?}).$$

⁵The latter is necessary because it can be the case that $\mathbf{r}_{x,y} \in v$, but either $\text{aw}_x \notin v$ or $\text{aw}_y \notin v$, and hence $(x,y) \notin \mathcal{R}_v$.

If one prefers to capture the more specific notions of necessary and deductive completions (Def. 2), then it is necessary to check whether the corresponding additional constraint is satisfied after each execution of $\text{mkComp}^{\text{pla}}(\mathcal{IB})$ and the computation of the corresponding extensions. However, we have to be cautious here, because the execution of $\text{mkComp}^{\text{pla}}(\mathcal{IB})$ may make true some support variables corresponding to elements of $\mathcal{S}^?$. An easy solution to this problem consists in making a copy of support variables before the execution of $\text{mkComp}^{\text{pla}}(\mathcal{IB})$. We use yet another set of fresh copies $\text{SUP}_{\mathbf{A} \times \mathbf{A}}'' = \{\mathbf{s}_{x,y}'' \mid (x,y) \in \mathbf{A} \times \mathbf{A}\}$ not to interfere with the later execution of $\text{deductive2necessary}$. So, the first part of our program for computing necessary and deductive completions will be

$$\text{copy}''(\text{SUP}_{\mathbf{A} \times \mathbf{A}}) = \mathop{\text{;}}_{x,y \in \mathbf{A}} (\text{if } \mathbf{s}_{x,y} \text{ then } +\mathbf{s}_{x,y}''; \text{else skip}).$$

Now, we can express the constraints corresponding to necessary and deductive completions: $\text{Constraint}^{\text{nec}} = \bigwedge_{x,y \in \mathbf{A}} ((\text{in}_x \wedge \mathbf{s}_{y,x}'') \rightarrow \text{in}_y)$ and $\text{Constraint}^{\text{ded}} = \bigwedge_{x,y \in \mathbf{A}} ((\text{in}_x \wedge \mathbf{s}_{x,y}'') \rightarrow \text{in}_y)$. And finally,

$$\text{mkComp}_{\sigma}^{\text{t}}(\mathcal{IB}) = \text{copy}''(\text{SUP}_{\mathbf{A} \times \mathbf{A}}); \text{mkComp}^{\text{pla}}(\mathcal{IB}); \\ ([\text{addAttacks}^{\text{t}}; \text{mkExt}^{\sigma}] \text{Constraint}^{\text{t}})?$$

Proposition 8 *Let $\mathcal{IB} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^?, \mathcal{S}, \mathcal{S}^? \rangle$, $\text{t} \in \{\text{pla}, \text{nec}, \text{ded}\}$, and let σ be a semantics. Then:*

- *If $\langle v_{\mathcal{IB}}, v \rangle \in \|\text{mkComp}_{\sigma}^{\text{t}}(\mathcal{IB})\|$, then $\langle \mathcal{A}_v, \mathcal{R}_v, \mathcal{S}_v \rangle \in \text{completions}_{\sigma}^{\text{t}}(\mathcal{IB})$.*
- *If $\langle \mathcal{A}_c, \mathcal{R}_c, \mathcal{S}_c \rangle \in \text{completions}_{\sigma}^{\text{t}}(\mathcal{IB})$, then there is a $v \subseteq \text{Prp}$ s.t. $v \cap \text{Prp}_{\mathbf{A}} = v_{\langle \mathcal{A}_c, \mathcal{R}_c, \mathcal{S}_c \rangle}$ and $\langle v_{\mathcal{IB}}, v \rangle \in \|\text{mkComp}_{\sigma}^{\text{t}}(\mathcal{IB})\|$.*

Argument acceptance for IBAFs in DL-PA. We have then all the ingredients to reduce argument acceptance problems of IBAFs to model checking problems in DL-PA.

Proposition 9 *Let $\mathcal{IB} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^?, \mathcal{S}, \mathcal{S}^? \rangle$, let $a \in \mathcal{A}$, let $\sigma \in \{\text{co}, \text{pr}, \text{gr}, \text{st}\}$, let $\text{t}_1 \in \{\text{pla}, \text{nec}, \text{ded}\}$ and $\text{t}_2 \in \{\text{nec}, \text{ded}\}$ s.t. if $\text{t}_1 \in \{\text{nec}, \text{ded}\}$, then $\text{t}_1 = \text{t}_2$. Then:*

- *The answer to σ^{t_2} - t_1 -PCA with input \mathcal{IB} and a is YES iff $v_{\mathcal{IB}} \models \langle \text{mkComp}_{\sigma}^{\text{t}_1}(\mathcal{IB}); \text{addAttacks}^{\text{t}_2}; \text{mkExt}^{\sigma} \rangle \text{in}_a$.*
- *The answer to σ^{t_2} - t_1 -NCA with input \mathcal{IB} and a is YES iff $v_{\mathcal{IB}} \models [\text{mkComp}_{\sigma}^{\text{t}_1}(\mathcal{IB})] \langle \text{addAttacks}^{\text{t}_2}; \text{mkExt}^{\sigma} \rangle \text{in}_a$.*
- *The answer to σ^{t_2} - t_1 -PSA with input \mathcal{IB} and a is YES iff $v_{\mathcal{IB}} \models \langle \text{mkComp}_{\sigma}^{\text{t}_1}(\mathcal{IB}) \rangle [\text{addAttacks}^{\text{t}_2}; \text{mkExt}^{\sigma}] \text{in}_a$.*
- *The answer to σ^{t_2} - t_1 -NSA with input \mathcal{IB} and a is YES iff $v_{\mathcal{IB}} \models [\text{mkComp}_{\sigma}^{\text{t}_1}(\mathcal{IB}); \text{addAttacks}^{\text{t}_2}; \text{mkExt}^{\sigma}] \text{in}_a$.*

6 Conclusion

This paper was devoted to the definition and the study of IBAFs: enriched frameworks for abstract argumentation taking into account two kinds of interaction between arguments (attacks and necessary or deductive supports) and considering at the same time that the elements (arguments or interactions) of this framework can be uncertain. Reasoning about IBAFs is done through the notion of completion: any uncertain element can be considered present or absent and so several “variants” of the IBAF may be built, each variant is called *completion* and corresponds to a classical BAF without uncertainty. Then, the semantics of IBAFs are defined by the application of the corresponding semantics on these completions. Due to the specificity of the support relations, two distinct approaches were proposed leading to two different definitions of completion: the first one corresponds to the notion of *conditionally certain support* whereas the second one introduces the notion of *unconditionally certain support*. Each kind of completion is in turn used for the definition of the semantics. Tight complexity results are given in the first case and upper bounds are presented in the second case. Finally, a logical encoding using DL-PA was proposed for both cases.

In terms of future works, several directions can be explored. First, we could complete this study for other IBAFs by taking into account some other support relations (for instance the evidential one). Second, the same work could be done for other enriched frameworks (for instance, those with higher-order interactions, or those with collective ones). A third interesting point could be to implement the computation of semantics of all these incomplete frameworks using DL-PA solvers and to make some experiments for evaluating our encoding and comparing with other approaches (perhaps with a more direct computation of semantics without using logics). Finally, a fourth line of future research consists in focusing on the uncertainty aspect of IBAFs by, e.g., considering a recursive form of uncertainty [34], or proposing a direct approach (*i.e.* without using completions) to define extension-based semantics in the style of [30].

Acknowledgments

We want to acknowledge our colleague Sylvie Doutre for the very rich exchanges about the topic of this paper.

References

- [1] Leila Amgoud and Claudette Cayrol. A reasoning model based on the production of acceptable arguments. *Ann. Math. Artif. Intell.*, 34(1-3):197–215, 2002.
- [2] Philippe Balbiani, Andreas Herzig, and Nicolas Troquard. Dynamic logic of propositional assignments: a well-behaved variant of pdl. In *Proc. of LICS'13*, pages 143–152, 2013.
- [3] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. Abstract argumentation frameworks and their semantics. In *Handbook of Formal Argumentation*, pages 159–236. College Publications, 2018.
- [4] Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. Extending abstract argumentation systems theory. *Artif. Intell.*, 120(2):251–270, 2000.
- [5] Howard Barringer, Dov Gabbay, and John Woods. Temporal dynamics of support and attack networks : From argumentation to zoology. In Dieter Hutter and Werner Stephan, editors, *Mechanizing Mathematical Reasoning, Essays in Honor of Jörg H. Siekmann. LNAI 2605*, pages 59–98. Springer, 2005.
- [6] Dorothea Baumeister, Matti Järvisalo, Daniel Neugebauer, Andreas Niskanen, and Jörg Rothe. Acceptance in incomplete argumentation frameworks. *Artif. Intell.*, 295:103470, 2021.
- [7] Dorothea Baumeister, Daniel Neugebauer, Jörg Rothe, and Hilmar Schadrack. Verification in incomplete argumentation frameworks. *Artif. Intell.*, 264:1–26, 2018.
- [8] Philippe Besnard and Sylvie Doutre. Checking the acceptability of a set of arguments. In *10th International Workshop on Non-Monotonic Reasoning (NMR 2004)*, pages 59–64, 2004.
- [9] Guido Boella, Dov M. Gabbay, Leendert W. N. van der Torre, and Serena Villata. Support in abstract argumentation. In *Proc. of COMMA '10*, pages 111–122, 2010.
- [10] Álvaro Carrera and Carlos A Iglesias. A systematic review of argumentation techniques for multi-agent systems research. *Artif. Intell. Rev.*, 44(4):509–535, 2015.
- [11] Claudette Cayrol, Andrea Cohen, and Marie-Christine Lagasquie-Schiex. Higher-order interactions (bipolar or not) in abstract argumentation: A state of the art. In *Handbook of Formal Argumentation, Volume 2*, pages 3–118. College Publications, 2021.
- [12] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In *Proc. of ECSQARU'05*, pages 378–389, 2005.

- [13] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Bipolarity in argumentation graphs: Towards a better understanding. *Int. J. Approx. Reason.*, 54(7):876–899, 2013.
- [14] Andrea Cohen, Alejandro Javier García, and Guillermo Ricardo Simari. Backing and undercutting in abstract argumentation frameworks. In *Proc. of FoIKS'12*, pages 107–123, 2012.
- [15] Sylvie Coste-Marquis, Caroline Devred, Sébastien Konieczny, Marie-Christine Lagasquie-Schiex, and Pierre Marquis. On the merging of Dung’s argumentation systems. *Artif. Intell.*, 171(10-15):730–753, 2007.
- [16] Kristijonas Cyras, Claudia Schulz, and Francesca Toni. Capturing bipolar argumentation in non-flat assumption-based argumentation. In *Proc. of PRIMA'17*, pages 386–402, 2017.
- [17] Sylvie Doutre, Andreas Herzig, and Laurent Perrussel. A dynamic logic framework for abstract argumentation. In *Proc. of KR'14*, 2014.
- [18] Sylvie Doutre, Andreas Herzig, and Laurent Perrussel. Abstract argumentation in dynamic logic: Representation, reasoning and change. In *Proc. of CLAR'18*, pages 153–185, 2018.
- [19] Sylvie Doutre, Faustine Maffre, and Peter McBurney. A dynamic logic framework for abstract argumentation: adding and removing arguments. In *Proc. of IEA/AIE'17*, pages 295–305, 2017.
- [20] 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.
- [21] Bettina Fazzinga, Sergio Flesca, and Filippo Furfaro. Probabilistic bipolar abstract argumentation frameworks: complexity results. In *Proc. of IJCAI'18*, pages 1803–1809, 2018.
- [22] Bettina Fazzinga, Sergio Flesca, and Filippo Furfaro. Reasoning over argument-incomplete AAFs in the presence of correlations. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 189–195. ijcai.org, 2021.
- [23] Bettina Fazzinga, Sergio Flesca, and Filippo Furfaro. Reasoning over attack-incomplete AAFs in the presence of correlations. In Meghyn Bienvenu, Gerhard Lakemeyer, and Esra Erdem, editors, *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning, KR 2021, Online event, November 3-12, 2021*, pages 301–311, 2021.

- [24] Anis Gargouri, Sébastien Konieczny, Pierre Marquis, and Srdjan Vesic. On a notion of monotonic support for bipolar argumentation frameworks. In *Proc. of AAMAS'21*, pages 546–554, 2021.
- [25] Andreas Herzig and Antonio Yuste-Ginel. Abstract argumentation with qualitative uncertainty: An analysis in dynamic logic. In *Proc. of CLAR'21*, pages 190–208. Springer, 2021.
- [26] Anthony Hunter, Sylwia Polberg, Nico Potyka, Tjitze Rienstra, and Matthias Thimm. Probabilistic argumentation: A survey. *Handbook of Formal Argumentation*, 2:397–441, 2021.
- [27] Nikos Karacapilidis and Dimitris Papadias. Computer supported argumentation and collaborative decision making: the HERMES system. *Inf. Syst.*, 26(4):259–277, 2001.
- [28] Amin Karamlou, Kristijonas Cyras, and Francesca Toni. Complexity results and algorithms for bipolar argumentation. In *Proc. of AAMAS'19*, pages 1713–1721, 2019.
- [29] Jean-Guy Mailly. Constrained incomplete argumentation frameworks. In Jirina Vejnárová and Nic Wilson, editors, *Proc. of ECSQARU'21*, volume 12897 of *Lecture Notes in Computer Science*, pages 103–116. Springer, 2021.
- [30] Jean-Guy Mailly. Extension-based semantics for incomplete argumentation frameworks. In *Proc. of CLAR'21*, pages 322–341, 2021.
- [31] Jean-Guy Mailly. Yes, no, maybe, I don't know: Complexity and application of abstract argumentation with incomplete knowledge. *Argument Comput.*, 13(3):291–324, 2022.
- [32] Farid Nouioua and Vincent Risch. Argumentation frameworks with necessities. In *Proc. of SUM'11*, volume 6929, pages 163–176, 2011.
- [33] Nir Oren and Timothy J. Norman. Semantics for evidence-based argumentation. In *Proc. of COMMA'08*, volume 172, pages 276–284. IOS Press, 2008.
- [34] Tjitze Rienstra, Matthias Thimm, and Nir Oren. Opponent models with uncertainty for strategic argumentation. In *Proc. of IJCAI'13*, 2013.
- [35] Antonio Yuste-Ginel and Andreas Herzig. Qualitative uncertainty and dynamics of argumentation through dynamic logic. *J. Log. Comput.*, 2023. exac098.

A Proofs

A.1 Proofs for Section 4

Recall that all complexity lower bounds are provided by the known results on Incomplete Argumentation Frameworks, as summarized in Table 1

σ	σ -PCA	σ -NCA	σ -PSA	σ -NSA
ad	NP-c	Π_2^P -c	trivial	trivial
st	NP-c	Π_2^P -c	Σ_2^P -c	coNP-c
co	NP-c	Π_2^P -c	NP-c	coNP-c
gr	NP-c	coNP-c	NP-c	coNP-c
pr	NP-c	Π_2^P -c	Σ_3^P -c	Π_2^P -c

Table 1: Complexity of acceptability for IAFs. PCA (resp. PSA) and NCA (resp. NSA) means Possible Credulous (resp. Skeptical) Acceptability and Necessary Credulous (resp. Skeptical) Acceptability [6, 31]

Hence, in what follows, we focus on identifying upper bounds:

[**Proposition 1**]

Proof: For $\mathbf{t} \in \{\text{nec}, \text{ded}\}$, the problem is solved by the simple non-deterministic Algorithm 1. Depending on the complexity of verifying an extension of a BAF, this algorithm is an NP algorithm if verification is in P, *i.e.* for $\sigma^{\mathbf{t}} \in \{\text{ad}^{\mathbf{t}}, \text{gr}^{\mathbf{t}}, \text{st}^{\mathbf{t}}, \text{co}^{\mathbf{t}}\}$. Finally, since $\text{pr}^{\mathbf{t}}$ -extensions are \subseteq -maximal $\text{ad}^{\mathbf{t}}$ -sets, then $\text{ad}^{\mathbf{t}}$ -PCA is equivalent to $\text{pr}^{\mathbf{t}}$ -PCA, hence the result for this last semantics. Without this particular case, Algorithm 1 would have only provided a Σ_2^P upper bound for $\text{pr}^{\mathbf{t}}$ -PCA. \square

Algorithm 1 Solve PCA

Input: $\mathcal{IB} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^?, \mathcal{S}, \mathcal{S}^? \rangle$, $a \in \mathcal{A} \cup \mathcal{A}^?$, σ a semantics, $\mathbf{t} \in \{\text{nec}, \text{ded}\}$.

- 1: Non-deterministically guess $\mathcal{B}^* = \langle \mathcal{A}^*, \mathcal{R}^*, \mathcal{S}^* \rangle$
 - 2: Non-deterministically guess $E \subseteq \mathcal{A}^*$ s.t. $a \in E$
 - 3: **if** \mathcal{B}^* is a pla-completion of \mathcal{IB} . **then**
 - 4: **if** E is a $\sigma^{\mathbf{t}}$ -extension of \mathcal{B}^* **then**
 - 5: **return** YES
 - 6: **end if**
 - 7: **end if**
 - 8: **return** NO
-

[**Proposition 2**]

Proof: We assume a fixed $\mathbf{t} \in \{\text{nec}, \text{ded}\}$, and we focus on $\sigma^{\mathbf{t}}\text{-NSA}$, the complementary problem of $\sigma^{\mathbf{t}}\text{-NSA}$. This problem is solved by Algorithm 2. With the same reasoning as for the previous proof, this algorithm provides a NP upper bound for $\sigma^{\mathbf{t}}\text{-NSA}$ for $\sigma^{\mathbf{t}} \in \{\text{gr}^{\mathbf{t}}, \text{st}^{\mathbf{t}}, \text{co}^{\mathbf{t}}\}$, and Σ_2^P for $\sigma^{\mathbf{t}} = \text{pr}^{\mathbf{t}}$, and we can conclude with the results for $\sigma^{\mathbf{t}}\text{-NSA}$. \square

Algorithm 2 Solve coNSA

Input: $\mathcal{IB} = \langle \mathcal{A}, \mathcal{A}^?, \mathcal{R}, \mathcal{R}^?, \mathcal{S}, \mathcal{S}^? \rangle$, $a \in \mathcal{A} \cup \mathcal{A}^?$, σ a semantics, $t \in \{\text{nec}, \text{ded}\}$.

- 1: Non-deterministically guess $\mathcal{B}^* = \langle \mathcal{A}^*, \mathcal{R}^*, \mathcal{S}^* \rangle$
- 2: Non-deterministically guess $E \subseteq \mathcal{A}^*$ s.t. $a \notin E$
- 3: **if** \mathcal{B}^* is a pla-completion of \mathcal{IB} . **then**
- 4: **if** E is a σ^t -extension of \mathcal{B}^* **then**
- 5: **return** NO
- 6: **end if**
- 7: **end if**
- 8: **return** YES

For the following proof, we need to refer to the complexity of reasoning with BAFs, which is summarized in Table 2

Semantics σ	σ -Ver	σ -Cred	σ -Skep	σ -Exist	σ -NE
ad	P	NP-c	Trivial	Trivial	NP-c
gr	P	P	P	Trivial	P
st	P	NP-c	coNP-c	NP-c	NP-c
co	P	NP-c	P	Trivial	P
pr	coNP-c	NP-c	Π_2^P -c	Trivial	NP-c

Table 2: Complexity of reasoning with BAFs. Results for ad, pr and st semantics come from [16, 28]. Complexity of σ -Ver for co and gr semantics is given in [21].

New results are also given (in grey cells) due to the fact that building the new attack relation following the necessary or deductive meaning can obviously be made in polynomial time w.r.t. the number of arguments.

Now we can continue with the proofs.

[Proposition 3]

Proof: The proof follows a similar approach as the previous ones, focusing on the complement problem of σ -NCA. Non-deterministically guess a BAF \mathcal{B}^* and check in polynomial time whether it is a completion of \mathcal{IB} for the given meaning of supports. Then, check whether the given argument a is credulously accepted in \mathcal{B}^* . This yields a NP algorithm (when credulous acceptability is polynomial) or a Σ_2^P algorithm (when credulous acceptability is in NP) for the complement of σ -NCA, hence the result. \square

[Proposition 4]

Proof: Non-deterministically guess \mathcal{B}^* and check in polynomial time whether it is a completion of \mathcal{IB} . Then simply check whether the given argument a is skeptically accepted in \mathcal{B}^* . The complexity of skeptical acceptability (P for {gr, co}, NP for st, and Π_2^P for pr) induces the results. \square

Now we focus on the cases where completions are computed using the necessary or deductive meaning.

[Proposition 5]

Proof: We can compute the grounded semantics of \mathcal{B}^* in polynomial time and check (still in polynomial time) for each $(a, b) \in \mathcal{S}^*$ whether $a \in \text{gr}(\mathcal{B}^*)$ and $b \in \text{gr}(\mathcal{B}^*)$. \square

[Proposition 6]

Proof: For $\sigma \in \{\text{ad}, \text{co}, \text{st}\}$, we can build in polynomial time (following *e.g.* the approach from [8]) a propositional formula $\Phi_\sigma^{\mathcal{B}^*}$ s.t. $\text{mod}(\Phi_\sigma^{\mathcal{B}^*}) = \sigma(\mathcal{B}^*)$. Then, for each $(a, b) \in \mathcal{S}^*$ we can check in NP whether $\Phi_\sigma^{\mathcal{B}^*} \wedge (b \rightarrow a)$ (for nec) or $\Phi_\sigma^{\mathcal{B}^*} \wedge (a \rightarrow b)$ (for ded) is satisfiable. Since there is a polynomially large number of supports in \mathcal{S} , checking whether \mathcal{B}^* is a nec/ded completion is a $\text{P}^{\text{NP}} = \Delta_2^{\text{P}}$ problem. \square

A.2 Proofs for Section 5

[Proposition 7]

Proof: As preliminary steps, one has to show the correctness of the involved programs and formulas. In particular, let $v \subseteq \text{Prp}$ be a valuation and let $\langle \mathcal{A}_v, \mathcal{R}_v, \mathcal{S}_v \rangle$ be the BAF represented by v , it holds that:

- If $(v, v') \in \|\text{transClosure}\|$, then $\mathcal{S}_v^+ = \mathcal{S}_{v'}$.
- $v \models (\mathbf{r}_{x,y}^{\text{aw}} \vee \mathbf{r}_{x,y}^{\text{case1}} \vee \mathbf{r}_{x,y}^{\text{case2}})$ iff $(x, y) \in \mathcal{R}_v^{\text{nec}}$.
- If $(v, v') \in \|\text{addAttacks}^{\text{nec}}\|$, then $\mathcal{R}_v^{\text{nec}} = \mathcal{R}_{v'}$.
- If $(v, v') \in \|\text{deductive2necessary}\|$, then $\mathcal{R}_v^{-1} = \mathcal{R}_{v'}$.
- If $(v, v') \in \|\text{addAttacks}^{\text{ded}}\|$, then $\mathcal{R}_v^{\text{ded}} = \mathcal{R}_{v'}$.

The proof of these lemmas are left to the reader. Then, the claim of the proposition follows from the last three bullet points, and the correctness of mkExt for our target semantics $\{\text{co}, \text{st}, \text{gr}, \text{pr}\}$ (see, *e.g.*, [35]). \square

[Proposition 8]

Proof: We prove both items for each possible value of t .

[$t = \text{pla}$] For the first item, suppose $\langle v_{\mathcal{IB}}, v \rangle \in \|\text{mkComp}^{\text{pla}}(\mathcal{IB})\|$. By the definition of $\text{mkComp}^{\text{pla}}(\mathcal{IB})$ and the meaning of mkTrueSome , we have that $v = v_{\mathcal{IB}} \cup \text{AW}_{\mathcal{A}'} \cup \text{ATT}_{\mathcal{R}'}$ with $\mathcal{A}' \subseteq \mathcal{A}^?$, $\mathcal{R}' \subseteq \mathcal{R}^?$ and $\mathcal{S}' \subseteq \mathcal{S}^?$. The latter implies, by definition of $(\cdot)_v$ that $\mathcal{A}_v = \mathcal{A} \cup \mathcal{A}'$, $\mathcal{R}_v = \mathcal{R} \cup \mathcal{R}'$ and $\mathcal{S}_v = \mathcal{S} \cup \mathcal{S}'$. The last two claims imply, by Definition 2, that $\langle \mathcal{A}_v, \mathcal{R}_v, \mathcal{S}_v \rangle \in \text{completions}^{\text{pla}}(\mathcal{IB})$.

For the second item, the valuation that satisfies the existential claim is simply $v_{\langle \mathcal{A}_c, \mathcal{R}_v, \mathcal{S}_c \rangle}$. Let us show it. Suppose that $\langle \mathcal{A}_c, \mathcal{R}_v, \mathcal{S}_c \rangle \in \text{completions}^{\text{pla}}(\mathcal{IB})$, which amounts, by Definition 2, to $\mathcal{A} \subseteq \mathcal{A}_c \subseteq \mathcal{A} \cup \mathcal{A}^?$; $\mathcal{R} \cap (\mathcal{A}_c \times \mathcal{A}_c) \subseteq \mathcal{R}_c \subseteq (\mathcal{R} \cup \mathcal{R}^?) \cap (\mathcal{A}_c \times \mathcal{A}_c)$ and $\mathcal{S} \cap (\mathcal{A}_c \times \mathcal{A}_c) \subseteq \mathcal{S}_c \subseteq (\mathcal{S} \cup \mathcal{S}^?) \cap (\mathcal{A}_c \times \mathcal{A}_c)$. Now, recall that $v_{\mathcal{IB}} = v_{\langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle}$. From the two previous statements we can deduce that the set of variables whose value differs from $v_{\mathcal{IB}}$ to $v_{\langle \mathcal{A}_c, \mathcal{R}_v, \mathcal{S}_c \rangle}$ must be a subset of $\text{AW}_{\mathcal{A}^?} \cup \text{ATT}_{\mathcal{R}^?} \cup \text{SUP}_{\mathcal{S}^?}$.

The latter implies, by the definition of $\text{mkComp}^{\text{pla}}$ and the meaning of mkTrueSome that $\langle v_{\mathcal{I}\mathcal{B}}, v_{\langle \mathcal{A}_c, \mathcal{R}_v, \mathcal{S}_c \rangle} \rangle \in \|\text{mkComp}^{\text{pla}}(\mathcal{I}\mathcal{B})\|$.

[$\mathbf{t} = \text{nec}$] **For the first item**, suppose $\langle v_{\mathcal{I}\mathcal{B}}, v \rangle \in \|\text{mkComp}_\sigma^{\text{nec}}(\mathcal{I}\mathcal{B})\|$, which amounts by definition of $\text{mkComp}^{\text{nec}}$ and the semantics of the sequential composition operator to $\langle v_{\mathcal{I}\mathcal{B}}, v \rangle \in \|\text{copy}''(\text{SUP}_{\mathbf{A} \times \mathbf{A}})\| \circ \|\text{mkComp}^{\text{pla}}(\mathcal{I}\mathcal{B})\| \circ \|([\text{addAttacks}^{\text{nec}}; \text{mkExt}^\sigma] \text{Constraint}^{\text{nec}})?\|$. That means that there are $v_1, v_2 \subseteq \text{Prp}$ s.t.

- (1) $\langle v_{\mathcal{I}\mathcal{B}}, v_1 \rangle \in \|\text{copy}''(\text{SUP}_{\mathbf{A} \times \mathbf{A}})\|$,
- (2) $\langle v_1, v_2 \rangle \in \|\text{mkComp}^{\text{pla}}(\mathcal{I}\mathcal{B})\|$, and
- (3) $\langle v_2, v \rangle \in \|([\text{addAttacks}^\dagger; \text{mkExt}^\sigma] \text{Constraint}^{\text{nec}})?\|$.

On the one hand, and since the execution of $\text{copy}''(\text{SUP}_{\mathbf{A} \times \mathbf{A}})$ does not alter the value of any variable from $\text{Prp}_{\mathbf{A}}$, we can deduce from (1), (2) and the previous case of this proposition that $\langle \mathcal{A}_{v_2}, \mathcal{R}_{v_2}, \mathcal{S}_{v_2} \rangle \in \text{completions}^{\text{pla}}(\mathcal{I}\mathcal{B})$. Moreover, note that by the semantics of ‘?’, we can deduce from (3) that $v_2 = v$. Substituting identical terms we have that $\langle \mathcal{A}_v, \mathcal{R}_v, \mathcal{S}_v \rangle \in \text{completions}^{\text{pla}}(\mathcal{I}\mathcal{B})$. So we’ve shown that the first part of the definition of necessary completion is satisfied by $\langle \mathcal{A}_v, \mathcal{R}_v, \mathcal{S}_v \rangle$.

On the other hand, suppose that $E \in \sigma^{\text{nec}}(\langle \mathcal{A}_v, \mathcal{R}_v, \mathcal{S}_v \rangle)$. The latter implies, by Proposition 7, that $E = E_{v_4}$ for some $\langle v, v_4 \rangle \in \|\text{addAttacks}^{\text{nec}}; \text{mkExt}^\sigma\|$. Note that from (3), the truth clause of $[\pi]$, and the previous assertion we can deduce that $v_4 \models \text{Constraint}^{\text{nec}}$. Suppose that $(x, y) \in \mathcal{S}$ (recall that \mathcal{S} is the set of certain supports), which implies $(x, y) \in \mathcal{S}_{v_{\mathcal{I}\mathcal{B}}}$, which in turn implies, by the definition of copy'' and the rest of involved programs, that $\mathbf{s}''_{x,y} \in v_4$. Further, suppose that $y \in E = E_{v_4}$. Both facts, together with $v_4 \models \text{Constraint}^{\text{nec}}$ imply that $x \in E_{v_4} = E$. So we’ve shown that the second part of the definition of necessary completion is satisfied by $\langle \mathcal{A}_v, \mathcal{R}_v, \mathcal{S}_v \rangle$. We can then assert that $\langle \mathcal{A}_v, \mathcal{R}_v, \mathcal{S}_v \rangle \in \text{completions}^{\text{nec}}(\mathcal{I}\mathcal{B})$.

For the second item, the valuation witnessing the existential claim is $v = v_{\langle \mathcal{A}_c, \mathcal{R}_c, \mathcal{S}_c \rangle} \cup \{\mathbf{s}''_{x,y} \mid (x, y) \in \mathcal{S}\}$. Let us show it. Suppose that $\langle \mathcal{A}_c, \mathcal{R}_c, \mathcal{S}_c \rangle \in \text{completions}_\sigma^{\text{nec}}(\mathcal{I}\mathcal{B})$, which by Def. 2 amounts to:

- (1) $\langle \mathcal{A}_c, \mathcal{R}_c, \mathcal{S}_c \rangle \in \text{completions}_\sigma^{\text{pla}}(\mathcal{I}\mathcal{B})$; and
- (2) $\forall E \in \sigma^{\text{nec}}(\langle \mathcal{A}_c, \mathcal{R}_c, \mathcal{S}_c \rangle), \forall (x, y) \in \mathcal{S}$ if $y \in E$, then $x \in E$.

Now, let $v_1 = v_{\mathcal{I}\mathcal{B}} \cup \{\mathbf{s}''_{x,y} \mid (x, y) \in \mathcal{S}\}$, we have that $\langle v_{\mathcal{I}\mathcal{B}}, v_1 \rangle \in \|\text{copy}''(\text{IN}_{\mathbf{A}})\|$ by definition of copy . Moreover, let $v_2 = v_{\langle \mathcal{A}_c, \mathcal{R}_c, \mathcal{S}_c \rangle} \cup \{\mathbf{s}''_{x,y} \mid (x, y) \in \mathcal{S}\}$, we can deduce that $\langle v_1, v_2 \rangle \in \|\text{mkComp}^{\text{pla}}(\mathcal{I}\mathcal{B})\|$ from (1), the previous case of this proof (i.e., $\mathbf{t} = \text{pla}$), and the fact that $\text{mkComp}^{\text{pla}}$ does not alter the value of $\mathbf{s}''_{x,y}$ -variables. Finally, note that v_2 survives to the test $([\text{addAttacks}^{\text{nec}}; \text{mkExt}^\sigma] \text{Constraint}^{\text{nec}})?$ because of (2) and Proposition 7.

[t = ded] This case is very similar to the previous one. The only important detail is the use of fresh variables $\{s''_{x,y} \mid (x,y) \in \mathbf{A} \times \mathbf{A}\}$ instead of $\{s'_{x,y} \mid (x,y) \in \mathbf{A} \times \mathbf{A}\}$ by the program `copy''`, so that the execution of `deductive2necessary` does not affect the evaluation of `Constraintded`. \square

[Proposition 9]

Proof: All cases follow similar lines of reasoning, using propositions 7 and 8 in the crucial steps. More precisely, some of the cases need a stronger version of Proposition 7 that is easily derivable from the meaning of the involved DL-PA programs. Namely, that the claim works not only for v_B but for any other valuation that agrees with it in the value given to variables in $\text{Prp} \cup \text{ATT}'_{\mathbf{A} \times \mathbf{A}} \cup \text{SUP}'_{\mathbf{A} \times \mathbf{A}}$ where $\text{ATT}'_{\mathbf{A} \times \mathbf{A}} = \{r'_{x,y} \mid (x,y) \in \mathbf{A} \times \mathbf{A}\}$, and the same for supports. More formally, we have that:

$$\begin{aligned} \sigma^t(\mathcal{B}) = & \{E_{v'} \mid (v, v') \in \|\text{addAttacks}^t; \text{mkExt}^\sigma\|, \\ & v \cap (\text{Prp}_{\mathbf{A}} \cup \text{ATT}'_{\mathbf{A} \times \mathbf{A}} \cup \text{SUP}'_{\mathbf{A} \times \mathbf{A}}) = v_B\}. \end{aligned} \quad (1)$$

Let us just see the proof for the first reasoning problem as an illustration:

[σ^{t_2} -t₁-PCA] **From-left-to-right.** Suppose that the answer to the reasoning problem is YES. The latter is equivalent, by definition of σ^{t_2} -t₁-PCA, to:

$\exists \mathcal{B} \in \text{completions}_{\sigma}^{t_1}(\mathcal{I}\mathcal{B}), \exists E \in \sigma^{t_2}(\mathcal{B})$ s.t. $a \in E$. This implies, by the proof of the second bullet of Proposition 8 that:

$\exists \mathcal{B}$ a BAF s.t. the valuation $v = v_B \cup \{s''_{x,y} \mid (x,y) \in \mathcal{S}\}$ satisfies $\langle v_{\mathcal{I}\mathcal{B}}, v \rangle \in \|\text{mkComp}_{\sigma}^{t_1}(\mathcal{I}\mathcal{B})\|, \exists E \in \sigma^{t_2}(\mathcal{B}), a \in E$. From this statement and (1) we can deduce that:

$\exists v \subseteq \text{Prp}$ s.t. $\langle v_{\mathcal{I}\mathcal{B}}, v \rangle \in \|\text{mkComp}_{\sigma}^{t_1}(\mathcal{I}\mathcal{B})\|, \exists v' \subseteq \text{Prp}, \langle v, v' \rangle \in \|\text{addAttacks}^{t_2}; \text{mkExt}^\sigma\|$ with $v' \models \text{in}_a$. This implies, by the semantics of $\langle \pi \rangle$ and the semantics of $;$ that:

$$v_{\mathcal{I}\mathcal{B}} \models \langle \text{mkComp}_{\sigma}^{t_1}(\mathcal{I}\mathcal{B}); \text{addAttacks}^{t_2}; \text{mkExt}^\sigma \rangle \text{in}_a.$$

From-right-to-left. Suppose that

$v_{\mathcal{I}\mathcal{B}} \models \langle \text{mkComp}_{\sigma}^{t_1}(\mathcal{I}\mathcal{B}); \text{addAttacks}^{t_2}; \text{mkExt}^\sigma \rangle \text{in}_a$. This implies, by the meaning of $\langle \pi \rangle$ and $;$ that:

$\exists v \subseteq \text{Prp}$ s.t. $\langle v_{\mathcal{I}\mathcal{B}}, v \rangle \in \|\text{mkComp}_{\sigma}^{t_1}(\mathcal{I}\mathcal{B})\|, \exists v' \subseteq \text{Prp}, \langle v, v' \rangle \in \|\text{addAttacks}^{t_2}; \text{mkExt}^\sigma\|$ with $a \in E_{v'}$. This implies by the first bullet of Proposition 8 that:

$\exists v \subseteq \text{Prp}$ s.t. $\langle \mathcal{A}_v, \mathcal{R}_v, \mathcal{S}_v \rangle \in \text{completions}_{\sigma}^{t_1}(\mathcal{I}\mathcal{B}), \exists v' \subseteq \text{Prp}, \langle v, v' \rangle \in \|\text{addAttacks}^{t_2}; \text{mkExt}^\sigma\|$ with $a \in E_{v'}$. This implies by Proposition 7 that:

$\exists v \subseteq \text{Prp}$ s.t. $\langle \mathcal{A}_v, \mathcal{R}_v, \mathcal{S}_v \rangle \in \text{completions}_{\sigma}^{t_1}(\mathcal{I}\mathcal{B}), \exists v' \subseteq \text{Prp}, E_{v'} \in \sigma^{t_2}(\langle \mathcal{A}_v, \mathcal{R}_v, \mathcal{S}_v \rangle)$ and $a \in E_{v'}$. This implies by definition of σ^{t_2} -t₁-PCA that:

The answer to σ^{t_2} -t₁-PCA is YES. \square