Highlights

Computational Complexity of Flat and Generic Assumption-Based Argumentation, with and without Probabilities
Kristijonas Čyraš, Quentin Heinrich, Francesca Toni

- We provide a comprehensive picture of the computational complexity of Assumption-Based Argumentation (ABA) for classical argumentation decision problems under several ABA semantics.
- We establish the computational complexity of probabilistic verification, credulous and sceptical acceptance function problems in Probabilistic ABA (PABA) under several argumentation semantics to be in the class $\text{FP}^{\#P}$. 
Computational Complexity of Flat and Generic Assumption-Based Argumentation, with and without Probabilities

Kristijonas Ėras*, Quentin Heinrich, Francesca Toni

Imperial College London, 180 Queen’s Gate, London SW7 2AZ, United Kingdom

Abstract

Reasoning with probabilistic information has recently attracted considerable attention in argumentation, and formalisms of Probabilistic Abstract Argumentation (PAA), Probabilistic Bipolar Argumentation (PBA) and Probabilistic Structured Argumentation (PSA) have been proposed. These foundational advances have been complemented with investigations on the complexity of some approaches to PAA and PBA, but not to PSA. We study the complexity of an existing form of PSA, namely Probabilistic Assumption-Based Argumentation (PABA), a powerful, implemented formalism which subsumes several forms of PAA and other forms of PSA. Specifically, we establish membership (general upper bounds) and completeness (instantiated lower bounds) of reasoning in PABA for the class $\text{FP}^{\#P}$ (of functions with a $\#P$-oracle for counting the solutions of an $NP$ problem) with respect to newly introduced probabilistic verification, credulous and sceptical acceptance function problems under several ABA semantics. As a by-product necessary to establish PABA complexity results, we provide a comprehensive picture of the ABA complexity landscape (for both flat and generic, possibly non-flat ABA) for the classical decision problems of verification, existence, credulous and sceptical acceptance under those ABA semantics.

Keywords: Complexity, Probabilistic Argumentation, Structured Argumentation, Assumption-based Argumentation

1. Introduction

Argumentation is a well-established reasoning paradigm, well-used and understood in multi-agent settings, see e.g. [1, 2, 3, 4], and with applications e.g. in legal (see [5] for an overview) and medical domains (see e.g. [6, 7, 8, 9, 10, 11, 12, 13, 14]). Probabilistic argumentation systems have been considered as ways to combine logic with...
probability [15] as well as to model uncertainty of logical arguments [16]. Reasoning with probabilistic evidence is also a pertinent issue in both medical [8] and legal [17] domains. For example, Dung and Thang suggest in [18] applying probabilistic argumentation to jury-based adjudication. Recent empirical evaluations support the need for probabilistic argumentation [19] in general.

Several forms of probabilistic argumentation have been proposed, following one or another of two approaches [20, 21]: the constellations approach, e.g. [18, 22, 23, 24, 25, 26, 27], assuming uncertainty over the topology of argumentation frameworks; and the epistemic approach, e.g. [28, 20], assuming uncertainty over belief in arguments. In this paper we focus on a formalism following the first approach: Probabilistic Assumption-Based Argumentation (PABA), a form of Probabilistic Structured Argumentation (PSA). The formalism was originally proposed by Dung and Thang in [18] to incorporate probabilistic reasoning within Assumption-Based Argumentation (ABA [29]), a structured argumentation reasoning framework. PABA provides a methodology for determining the strength of arguments built from standard rules and assumptions, as in ABA, as well as probabilistic rules and assumptions. We illustrate PABA with a simple example as follows.

**Example 1.1.** Consider a clinical decision making problem where we aim to model the following information (loosely based on [13]):

- Two mutually incompatible actions of *administering an NSAID* (non-steroidal anti-inflammatory drug) and *not administering Aspirin* (which is a specific NSAID) are available for a given patient; call these actions $A$ and $A'$, respectively.
- Administering an NSAID generally has effect of decreasing blood coagulation (call this effect $E$), whereas not administering Aspirin would help to *avoid gastrointestinal bleeding* (call this effect $E'$), unless complications are present.

Given a patient, one can make assumption, call it $\text{no\_compl}$, regarding the absence of complications until further evidence is found, regarding the avoidance of gastrointestinal bleeding (effect $E'$) given that Aspirin is not administered (action $A'$). Further, the effect $E$ of decreasing blood coagulation can be made conditional on the probabilistic effectiveness, call it $\text{eff}$, of the administered NSAID (action $A$). This can be formalised in a PABA framework $\mathcal{F}_P = (\mathcal{A}_P, \mathcal{R}_P, \mathcal{F})$ with\(^2\)

- an ABA framework $\mathcal{F}$ containing
  - (standard) assumptions $\mathcal{A} = \{A, A', \text{no\_compl}\}$,
  - (standard) rules $\mathcal{R} = \{E \leftarrow A, \text{eff}, E' \leftarrow A', \text{no\_compl}\}$,
  - and contraries $\overline{A} = A'$, $\overline{A'} = A$ indicating that the two actions are incompatible,
- probabilistic assumptions $\mathcal{A}_P = \{\text{eff}\}$,
- and probabilistic rules $\mathcal{R}_P = \{[\text{eff} : 0.7] \leftarrow, \ [\neg\text{eff} : 0.3] \leftarrow\}$ representing the uncertainty about the effectiveness of $E$ assuming $A$.

\(^2\)Details on ABA will be given in Section 3.1, and on PABA in Section 5.2, where we also consider an extended version of this example.
In $\mathcal{F}_p$ both probabilistic and non-probabilistic arguments can be built, and we can determine how likely it is that a sentence (e.g. $E$ in this case) is accepted, meaning how likely it is that the patient’s blood coagulation will be decreased.

PABA relies on a probability distribution over worlds that amount to sets of probabilistic assumptions. PABA imposes no restrictions on probability assignments over probabilistic assumptions (in contrast to some approaches, such as [22, 26, 27], which assume independence in argumentation frameworks). PABA was shown by Hung to admit as instances several other probabilistic argumentation approaches [30], including forms of Probabilistic Abstract Argumentation (PAA) [18, 22] and another form of PSA, namely p-ASPIC [24], a probabilistic version of the ASPIC structured argumentation framework [31]. This shows PABA to be a powerful probabilistic argumentation formalism. Moreover, in order to support applications of PABA and its several instances, Hung advanced inference procedures as well as an implementation engine for PABA [30], particularly useful for the class of PABA frameworks that integrate Bayesian networks.

It is widely agreed that it is important to study computational complexity of probabilistic argumentation (see e.g. [21]). Arguably, the problems of set verification (denoted $\text{VER}$) as well as credulous and sceptical acceptance (denoted $\text{CA}$ and $\text{SA}$, respectively) under different semantics are the fundamental complexity problems in probabilistic and non-probabilistic argumentation alike [32, 33, 34, 21]. Indeed, they consider the computational costs of the key forms of reasoning (or inference) in argumentation frameworks: $\text{VER}$ asks how costly it is to determine if a given set of elements (e.g. arguments or assumptions) is an acceptable extension (i.e. a coherent point-of-view of the information represented) in a given argumentation framework; $\text{CA}/\text{SA}$ asks how costly it is to determine if a given element can/must be accepted (e.g. as true, or defensible, or believable, or probable).

Unfortunately, complexity results underpinning inference in PABA are lacking. In this paper we fill this gap by studying the complexity of PABA and establish results for several complexity problems, under various PABA semantics, thus providing theoretical foundations for the implementation efforts as well as reasoning problems beyond. In particular, our analysis provides guidance towards the choice of suitable instances of PABA and of its computational machinery for applications.

The complexity of (several problems in) various other forms of probabilistic argumentation, notably PAA and Probabilistic Bipolar Argumentation (PBA), has been studied recently. Specifically, the problem of probabilistic verification $P-\text{VER}$ was studied in [27] for PAA (under the approach of [22]), and in [35] for PBA. Recently, Fazzinga et al. provided a thorough exposition in on the complexity of $P-\text{VER}$ and probabilistic credulous acceptance ($P-\text{CA}$) in [21] of various forms of PAA under different semantics. However, to the best of our knowledge, the complexity of reasoning in PABA, or in other forms of PSA for that matter, has not been studied to date. Moreover, the problem of probabilistic sceptical acceptance ($P-\text{SA}$) has been mostly neglected in general, for PAA, PBA and PSA alike (with the exception of [21] for PAA where $P-\text{SA}$ is mentioned among future work directions). We fill these gaps by studying the complexity all three problems, namely $P-\text{VER}$, $P-\text{CA}$ and $P-\text{SA}$, for PABA under several prominent semantics. In particular, we consider the grounded extension semantics, as
considered in [18], as well as the ideal and preferred extension semantics, as considered in [30], and the admissible, complete and stable extension semantics, defined for non-probabilistic ABA in [29]. Although the existing works on PABA consider only rules and assumptions that constitute flat ABA frameworks (as defined in [29]), our analysis extends to generic (possibly non-flat) ABA frameworks and semantics thereof, namely well-founded extensions [29] (amounting to grounded extensions in the flat case) and set-stable extensions, generalised from the ones defined in [36] for a restricted form of non-flat ABA frameworks (and amounting to stable extensions in the flat case). Finally, our analysis also considers the semi-stable extension semantics, originally defined for abstract argumentation in [37] and for generic (possibly non-flat) ABA in [38]. Thus, as a by-product of our analysis, we consider PABA under additional semantics than originally envisaged, as well as for generic, possibly non-flat instances of PABA.

To establish results for these many problems and semantics, we must establish many new complexity results for decision problems in ABA, complementing the existing results, notably those in [32] but also some of [39]. We do this by studying the general upper bounds of generic ABA frameworks as well as for the particular class of flat ABA frameworks. We also study the lower bounds of two instances of ABA, obtained by restricting the forms of rules, assumptions (and their contraries) of general-purpose ABA. Specifically, we study under various semantics a flat ABA instance known as LP-ABA [29, 32], defined to capture Logic Programming (LP), and we define and study a new generic (i.e. possibly non-flat) ABA instance, that we call Horn-ABA, which uses Horn logic as ABA’s underlying deductive system and therefore effectively subsumes LP-ABA too. Table 1 depicts the landscape of flat and generic ABA, with and without probabilities, that we study, under the various semantics mentioned earlier.

<table>
<thead>
<tr>
<th>Non-probabilistic</th>
<th>Flat</th>
<th>Generic</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABA [29, 40]</td>
<td>LP-ABA [29, 32]</td>
<td>Horn-ABA (§4.2.2)</td>
</tr>
<tr>
<td>Probabilistic</td>
<td>PABA</td>
<td>Flat [18, 30] and Generic §5.2</td>
</tr>
<tr>
<td></td>
<td>Horn-PABA (§6)</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: The landscape of ABA with and without probabilities studied in this paper. Instances of (non-probabilistic) ABA are based on Logic Programming (LP) in the flat case, and Propositional Horn logic (Horn) in the generic (possibly non-flat) case. Instances of PABA, the probabilistic variant of ABA, are based on the Horn-ABA instance of ABA, in both flat and generic cases. Entries in bold pertain to new cases (defined where indicated); the ABA instance Horn-ABA is given in Section 4.2.2; PABA is generalised to incorporate non-flat ABA frameworks in Section 5.2; Horn-PABA is given in Section 6.

Our overall contributions to the computational complexity of argumentation are as follows. As a precursor for the complexity analysis of ABA and PABA, in Section 3.3, we disambiguate the standard definition of the sceptical acceptance problem $S\alpha$ in (non-probabilistic) argumentation, leading to counter-intuitive outcomes, by integrating $S\alpha$ with the (non-probabilistic) existence problem $E\alpha$ to yield intuitive outcomes, while preserving existing complexity results. Then, as the first major contribution we establish additional upper and lower bounds for ABA ranging classes $P$, $NP$ and $coNP$ in the flat case, and $P$, $NP$, $coNP$, $\Sigma^p_2$, $\Pi^p_2$ and $\Delta^p_3$ in the generic case (with an appropri-
ate derivation oracle for the upper bounds, and instantiations as required for the lower bounds), for set verification VER, existence EX, credulous acceptance CA and sceptical acceptance SA problems, under admissible, complete, preferred, stable, set-stable and well-founded/grounded semantics. As the second major contribution we define the probabilistic function problem counterparts P-VER, P-CA and P-SA in PABA and establish that their complexity upper bounds are in the class \( \text{FP}^{\#P} \) (function problems solvable by a deterministic Turing machine in polynomial time, with a \( \#P \)-oracle counting the number of accepting paths of a non-deterministic polynomial-time Turing machine), in both flat and generic cases, under admissible, complete, preferred, stable, set-stable and well-founded/grounded semantics. We further establish completeness for this class as lower bounds of Horn-PABA. We conclude our exposition by proposing and investigating the complexity of novel decision problems for PABA, namely StrongCA and StrongSA, concerning credulous and sceptical acceptance almost everywhere (i.e. with probability 1). Tables 2, 3 and 4 summarise the complexity results obtained in this paper, alongside results existing in the literature.

### Table 2: Summary of existing (as reviewed in Section 3.2) and new (in bold) upper bounds for classical complexity problems in ABA, as established in Section 4.

<table>
<thead>
<tr>
<th>sem</th>
<th>VER</th>
<th>EX</th>
<th>CA</th>
<th>SA</th>
<th>VER</th>
<th>EX</th>
<th>CA</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>adm</td>
<td>( P^{1D} )</td>
<td>trivial (Yes)</td>
<td>( NP^{1D} )</td>
<td>D</td>
<td>( \text{coNP}^{1D} )</td>
<td>( (\Sigma_2^D)^P )</td>
<td>( (\Sigma_2^D)^P )</td>
<td>( (\Pi_2^D)^P )</td>
</tr>
<tr>
<td>cpl</td>
<td>( P^{1D} )</td>
<td>trivial (Yes)</td>
<td>( NP^{1D} )</td>
<td>( P^{1D} )</td>
<td>( \text{DP}^{1D} )</td>
<td>( (\Sigma_2^D)^P )</td>
<td>( (\Sigma_2^D)^P )</td>
<td>( (\Pi_2^D)^P )</td>
</tr>
<tr>
<td>prf</td>
<td>( \text{coNP}^{1D} )</td>
<td>trivial (Yes)</td>
<td>( NP^{1D} )</td>
<td>( (\Pi_2^D)^P )</td>
<td>( (\Pi_2^D)^P )</td>
<td>( (\Sigma_2^D)^P )</td>
<td>( (\Sigma_2^D)^P )</td>
<td>( (\Pi_2^D)^P )</td>
</tr>
<tr>
<td>set-stb</td>
<td>( \text{P}^{1D} )</td>
<td>( NP^{1D} )</td>
<td>( \text{coNP}^{1D} )</td>
<td>( \text{P}^{1D} )</td>
<td>( \text{NP}^{1D} )</td>
<td>( \text{NP}^{1D} )</td>
<td>( \text{coNP}^{1D} )</td>
<td></td>
</tr>
<tr>
<td>grd/w-f</td>
<td>( \text{P}^{1D} )</td>
<td>trivial (Yes)</td>
<td>( \text{P}^{1D} )</td>
<td>( \text{P}^{1D} )</td>
<td>( (\Delta_2^P)^P )</td>
<td>( (\Delta_2^P)^P )</td>
<td>( (\Delta_2^P)^P )</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Summary of existing (as reviewed in Section 3.2) and new (in bold) upper bounds for classical complexity problems in ABA, as established in Section 4.

### Table 3: Summary of existing (as reviewed in Section 3.2) and new (in bold) lower and upper bounds for classical complexity problems in instantiated ABA, as established in Section 4.

<table>
<thead>
<tr>
<th>sem</th>
<th>LP-ABA (Flat)</th>
<th>Horn-ABA (Generic)</th>
</tr>
</thead>
<tbody>
<tr>
<td>adm</td>
<td>( P )</td>
<td>trivial (Yes)</td>
</tr>
<tr>
<td>cpl</td>
<td>( P )</td>
<td>trivial (Yes)</td>
</tr>
<tr>
<td>prf</td>
<td>( \text{coNP}-c )</td>
<td>trivial (Yes)</td>
</tr>
<tr>
<td>set-stb</td>
<td>( P )</td>
<td>( NP-c )</td>
</tr>
<tr>
<td>grd/w-f</td>
<td>( P )</td>
<td>trivial (Yes)</td>
</tr>
</tbody>
</table>

Table 3: Summary of existing (as reviewed in Section 3.2) and new (in bold) lower and upper bounds for classical complexity problems in instantiated ABA, as established in Section 4.
We organise the paper as follows. In Section 2 we give necessary background on the elements of complexity theory. We then study the complexity of ABA: in Section 3 we give ABA background, recall existing complexity results relevant for our purposes and state the complexity problems of interest; in Section 4 we establish complexity results for ABA. We then proceed to study the complexity of PABA: in Section 5 we give PABA background and introduce the complexity problems of interest; in Section 6 we establish complexity results for PABA and take on some further PABA complexity considerations. We review related works in Section 7 and conclude with Section 8.

2. Elements of Complexity Theory

We here give background on the elements of complexity theory. We first briefly review basics of complexity of decision problems and reductions, following [41, 39], and then recap counting complexity following [41, 42, 43].

2.1. Decision Problems

The Polynomial Hierarchy (PH) of (complexity) classes is defined as follows:

- \( \Delta_i^P = \Sigma_i^P = \Pi_i^P = P \),
- \( \Delta_i^{P+} = P^{\Sigma_i^P} \forall i \geq 0 \),
- \( \Sigma_i^{P+} = \text{NP}^{\Sigma_i^P} \forall i \geq 0 \),
- \( \Pi_i^{P+} = \text{coNP}^{\Sigma_i^P} \forall i \geq 0 \),

where \( \mathcal{C}^D \) denotes the class of decision problems that can be solved by an algorithm in class \( C \) making calls to an oracle in class \( D \), and \( P \) and \( \text{NP} \) respectively denote the classes of decision problems that can be solved by a deterministic (respectively non-deterministic) Turing machine running in polynomial time. Having either an oracle for a problem \( L \in \mathcal{D} \) or its complement problem \( \bar{L} \in \text{coD} \) is the same, so \( \mathcal{C}^D = \mathcal{C}^{\text{coD}} \). For \( \text{coD} \) the complement class of \( D \). We thus have \( \Delta_i^P = P \), \( \Sigma_i^P = \text{NP} \), \( \Pi_i^P = \text{coNP} \), \( \Delta_i^P = \text{P}^{\text{NP}} \), \( \Sigma_i^P = \text{NP}^{\text{NP}} \) and \( \Pi_i^P = \text{coNP}^{\text{NP}} \). In this paper, we equate PH (as a collection of complexity classes) to the class \( \bigcup_{i \geq 0} \Sigma_i^P \).

Another class of interest in this paper is DP, which consists of problems that can be solved by exactly one call to each of NP- and coNP-oracles (formally, \( L \in \text{DP} \) iff \( \exists L_1 \in \text{NP}, \ L_2 \in \text{coNP} \) such that \( L = L_1 \cap L_2 \)). We have \( \text{DP} \subseteq \Delta_2^P \), and so we note that making calls to a DP-oracle is equivalent to making calls to an NP-oracle. In particular, \( \text{NP}^{\text{DP}} = \text{NP}^{\text{NP}} = \Sigma_2^P \).
A common method to prove that a problem $L$ belongs to $NP$ is giving a polynomial time algorithm that solves $L$ by making a non-deterministic choice, called \textit{guessing}. For example, the \textsc{SAT} problem of deciding whether a boolean formula $\Phi$ is satisfiable is in $NP$, by virtue of the following algorithm.

1. Guess a truth assignment for variables in $\Phi$.
2. Check if this assignment makes $\Phi$ true.

Checking the assignment is polynomial, while guessing it is a non-deterministic choice. We will employ this method for several proofs in the paper.

Throughout the paper, we implicitly make the assumption that $P \neq NP$ when saying that one problem is harder than another one as suggested by them being in different (but as yet unproven to be distinct) hierarchically related classes. This assumption has no effect on the complexity results that we establish but gives a flavour of the relative complexity of the problems considered.

2.2. Reductions

We will classify new problems by comparing them to known problems using \textit{many-one reductions} and \textit{Turing reductions}, where, for problems $L_1, L_2$:

- $L_1$ is many-one reducible to $L_2$, denoted $L_1 \leq_m L_2$, iff there exists a polynomial-time computable function $f$ such that for all inputs $x$ to $L_1$, $x$ satisfies $L_1$ iff $f(x)$ satisfies $L_2$;
- $L_1$ is Turing reducible to $L_2$, denoted $L_1 \leq_T L_2$, iff $L_1$ can be solved in polynomial time by using a polynomial number of calls to an oracle for $L_2$, also denoted $L_1 \in P^{L_2}$.

The following implication holds: $L_1 \leq_m L_2 \Rightarrow L_1 \leq_T L_2$.

Given a class $C$, a problem $L$ is \textit{$C$-hard for reduction $r$} iff every problem in $C$ is $r$-reducible to $L$. If $L$ is in $C$ too, then $L$ is \textit{$C$-complete for reduction $r$}, also denoted $C$-$c$. For instance, \textsc{SAT} is $NP$-complete (for many-one reduction). Note that in this paper we do not consider reductions within class $P$, and hence do not use the notion of $P$-completeness.

2.3. Function Problems and Counting Complexity

We now give background on the complexity of function problems required for the complexity analysis of PABA following [41, 42, 43].

In PABA, (most of) the complexity problems will not be decision problems, but function problems: they will consist of computing the output of a function, such as a probability, rather than a binary ‘Yes’/‘No’ output. Function problems can be defined with reference to decision problems as follows. Given a decision problem $L$, the associated function problem $FL$ consists in finding for an input $x$ to $L$, a witness $y$ which proves that $x$ satisfies $L$, whenever such $y$ exists. In general, for a class $C$ of decision problems, $FC$ is the class of function problems associated to the decision problems in $C$. For instance, $FP$ is the class of function problems that can be solved by a deterministic Turing machine running in polynomial time.

A specific complexity class of interest in this paper is that of counting problems [42]. The complexity class $\#P$ is the class of function problems for counting the number of accepting paths of a non-deterministic polynomial-time Turing machine. For
instance, \#SAT is a standard problem in \#P which consists in counting the number of satisfying assignments of a boolean formula.

Regarding reductions (see Section 2.2) in PH and function classes within FPH (consisting of classes of function problems associated to the decision problems in classes within PH), many-one reduction is generally used. In function classes above FPH and for counting problems, Turing reduction is often used instead. For instance, \#SAT is \#P-c (for Turing reduction). In our analysis of the complexity of counting problems for PABA in Section 6, we will accordingly use Turing reductions.

Counting complexity is generalised as follows [43]: Given a C-oracle, \#PC is the class of counting problems computable by an algorithm counting the number of accepting paths of a non-deterministic polynomial-time Turing machine with a C-oracle (i.e. counting the number of accepting paths of an NP C Turing machine). Specifically for a complexity class C within PH, we denote \#C = \#PC. For example, \#NP is the class of counting problems computable by an algorithm counting the number of accepting paths of a non-deterministic polynomial-time Turing machine with an NP-oracle (i.e. counting the number of accepting paths of an NP NP Turing machine). In particular, \#PH = \bigcup_{i \geq 0} \#ΣP i is the class of counting problems computable by an algorithm counting the number of accepting paths of a non-deterministic polynomial-time Turing machine with a PH-oracle (more precisely, with an oracle in PH).

In Section 6 we will also use the following fact: FP#P = FP#PH [43].

3. ABA: Background, Existing Results and Problems of Interest

In this section we first give background on ABA as well as existing results on its complexity. We then revisit the existence problem and lay down the complexity problems of interest in ABA.

3.1. ABA Background

We base background on ABA on [29, 36, 40].

An ABA framework is a tuple \((L, R, A, −)\), where:

- \((L, R)\) is a deductive system with \(L\) a language and \(R\) a set of rules of the form \(b_0 \leftarrow b_1, \ldots, b_m\) with \(m \geq 0\) and \(b_i \in L\) for \(i \in \{0, \ldots, m\}\); given \(r = b_0 \leftarrow b_1, \ldots, b_m\):
  - \(b_0\) is referred to as the head of \(r\), denoted head\((r)\), and
  - \(\{b_1, \ldots, b_m\}\) is referred to as the body of \(r\), denoted body\((r)\);
  - if \(m = 0\), then \(r\) is said to have an empty body, and is written as \(b_0 \leftarrow\), referred to as a fact, and body\((r)\) = \(\emptyset\);
- \(A \subseteq L\) is a non-empty set of assumptions;
- \(− : A \rightarrow L\) is a total contrary mapping: for \(a \in A\), the \(L\)-sentence \(\overline{a}\) is referred to as the contrary of \(a\).

Unless stated otherwise, we assume a fixed but otherwise arbitrary ABA framework \(\mathcal{F} = (L, R, A, −)\). Throughout the paper we assume that \(\mathcal{F}\) and all its components are finite.

An argument for \(p \in L\) supported by \(S \subseteq A\) and \(R \subseteq R\), denoted by \(S \vdash_R p\), is a finite tree with:
• the root labelled by $p$;
• leaves labelled either by a special symbol $\top \notin \mathcal{L}$ or by assumptions, with $\mathcal{S}$ being the set of all such assumptions;
• the children of non-leaf nodes $q$ labelled by the elements of the body of some rule $r$ from $\mathcal{R}$ with head $q$, where
  – if $r$ has empty body, then the child is labelled by $\top$, and
  – $\mathcal{R}$ is the set of all such rules.

$\mathcal{S} \vdash p$ denotes $\mathcal{S} \vdash^R p$ for some $R \subseteq \mathcal{R}$.

A $\subseteq \mathcal{A}$ attacks $B \subseteq \mathcal{A}$ iff there is $A' \vdash b$, for some $A' \subseteq A$ and $b \in B$. With an abuse of notation, we say that $A$ attacks $a \in \mathcal{A}$ whenever $A$ attacks $\{a\}$.

To define ABA semantics, we use the following auxiliary notions, for $S \subseteq \mathcal{A}$:
• we define the conclusions operator $Cn$ by $Cn(S) = \{ \varphi \in \mathcal{L} : \exists A \vdash \varphi, A \subseteq S \}$, and call $Cn(S)$ the conclusions of $S$, i.e. the set of sentences concluded by (arguments supported by subsets of) $S$;
• $S$ is closed iff $S = Cn(S) \cap \mathcal{A}$, i.e. $S$ contains all assumptions it concludes;
• $S$ is conflict-free iff $S$ does not attack itself;
• $S$ defends $a \in \mathcal{A}$ iff for all closed $B \subseteq \mathcal{A}$ attacking $a$, $S$ attacks $B$;
• we define the defence operator $Def$ by $Def(S) = \{ a \in \mathcal{A} : S$ defends $a \}$.

We will focus on the following ABA semantics. A set $S \subseteq \mathcal{A}$ of assumptions, also called an extension, is:
• admissible iff $S$ is closed, conflict-free and defends every $a \in S$ (i.e. $S \subseteq Def(S)$);
• complete iff $S$ is admissible and contains all assumptions it defends (i.e. $Def(S) \subseteq S$);
• preferred iff $S$ is $\subseteq$-maximally admissible;
• stable iff $S$ is closed, conflict-free and $S$ attacks every $b \in \mathcal{A} \setminus S$;
• set-stable iff $S$ is closed, conflict-free and attacks the closure $Cn(\mathcal{A}) \cap \mathcal{A}$ of every non-empty subset $\emptyset \neq A \subseteq \mathcal{A} \setminus S$;
• well-founded iff $S$ is the intersection of all complete extensions. (By convention, if no complete extension exists, then there is no well-founded extension.)

We say $\mathcal{F}$ is flat if every $S \subseteq \mathcal{A}$ is closed. In a flat ABA framework, stable and set-stable semantics coincide [36], and the well-founded extension is also called the grounded extension. By [29, Theorem 6.2], the grounded extension $\mathcal{G}$ is the least fix point of the operator $Def$, i.e. $\mathcal{G} = \bigcup_{0 \leq i \leq n} Def^i(\emptyset)$ with $n = |\mathcal{A}|$.

In what follows, we use the following obvious abbreviations to refer to ABA semantics: adm, cpl, prf, stb, set-stb, w-f, grd.

As observed in [40], most works on ABA focus exclusively on flat ABA frameworks. Flat ABA is indeed a powerful knowledge representation and reasoning formalism, capturing, for example, forms of LP, Default Logic [44] and other forms of decision making (see e.g. [29, 40]). However, since non-flat ABA allows for beliefs as assumptions to be deduced from other assumptions, it enables capturing of additional forms of reasoning, such as Autoepistemic Logic [45], Circumscription [46], forms of Bipolar Argumentation (see [36, 47]) and medical reasoning (see [13]). It is therefore important to study the complexity of both flat and general (possibly non-flat) ABA frameworks.
We consider the following illustration, in reference to Example 1.1 given in the Introduction.

**Example 3.1.** Inspired by a medical decision making problem formulated in ABA for reasoning with clinical guideline information (e.g. for chronic obstructive pulmonary disease, COPD [48]) as proposed in [12, 13], we aim to model the following information (abstracting away from the specific medical terminology, illustrated in Example 1.1):

- Two mutually incompatible actions $A$ and $A'$ are available for a given patient;
- Action $A$ generally has effect $E$, whereas action $A'$ leads to effect $E'$, unless complications are present (such as a patient’s disease exacerbation).

Given a patient, one may make assumptions regarding absence of complications until further evidence is found. This problem can be formalised in a flat ABA framework $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \top)$ thus:

- $\mathcal{A} = \{A, A', no\_compl\}$,
- $\mathcal{R} = \{E \leftarrow A, E' \leftarrow A', no\_compl\}$,
- $\overline{A} = A'$, $\overline{A'} = A$.

Here, $\{no\_compl\}$ is unattacked, while $\{A\}$ and $\{A'\}$ attack each other. Thus, both $\{no\_compl, A\}$ and $\{no\_compl, A'\}$ are stable, preferred, complete and admissible, whereas $S$ is the grounded extension of $\mathcal{F}$, and so complete and admissible too.

We invite the reader to consult e.g. [40] for extensive examples of flat and non-flat ABA frameworks, derivations, extensions under various semantics and other concepts.

### 3.2. Existing Complexity Results for ABA

We give some existing complexity results for ABA following [32, 39].

The following problems have been defined and analysed for ABA. For an ABA framework $\mathcal{F}$ and $\text{sem} \in \{\text{adm, prf, stb, cpl, grd}\}$:

- The set verification problem for a given $S \subseteq \mathcal{A}$, denoted $\text{VER}_{\text{sem}}(S)$, is the problem of deciding whether $S$ is a $\text{sem}$ extension of $\mathcal{F}$.
- The credulous acceptance problem for a given $\pi \in \mathcal{L}$, denoted $\text{CA}_{\text{sem}}(\pi)$, is the problem of deciding whether there exists a $\text{sem}$ extension $S \subseteq \mathcal{A}$ such that $\pi \in Cn(S)$.
- The sceptical acceptance problem for a given $\pi \in \mathcal{L}$, denoted $\text{SA}_{\text{sem}}(\pi)$, is the problem of deciding whether for each $\text{sem}$ extension $S \subseteq \mathcal{A}$ it holds that $\pi \in Cn(S)$.

Henceforth, $\text{VER}$, $\text{CA}$ and $\text{SA}$ abbreviate, respectively, the set verification, credulous acceptance and sceptical acceptance problems.

---

3By convention, $\mathcal{L}$ and (any unspecified part of) $\top$ are implicit from $\mathcal{A}$ and $\mathcal{R}$ as follows: unless $\pi$ appears in either $\mathcal{A}$ or $\mathcal{R}$, it is different from the sentences appearing in $\mathcal{A}$ or $\mathcal{R}$; thus, $\mathcal{L}$ consists of all the sentences appearing in $\mathcal{R}$, $\mathcal{A}$ and $\{\pi : a \in \mathcal{A}\}$. 

10
Example 3.2. In \( \mathcal{F} \) from Example 3.1, for instance, the answer to \( \text{CA}_{sem}(E) \) is ‘Yes’, and the answer to \( \text{SA}_{sem}(E') \) is ‘No’, for any \( sem \in \{ \text{prf}, \text{cpl}, \text{stb} \} \). The answers to both \( \text{CA}_{sem}(\text{no compl}) \) and \( \text{SA}_{sem}(\text{no compl}) \) are ‘Yes’, for any \( sem \in \{ \text{adm}, \text{prf}, \text{stb}, \text{cpl}, \text{grd} \} \). Note that \( \emptyset \) is admissible in \( \mathcal{F} \), so the answer to \( \text{SA}_{adm}(\text{no compl}) \) is ‘No’.

The complexity of these problems can be established only with respect to the complexity of the derivability in the underlying logic, i.e. the deductive system \( (\mathcal{L}, \mathcal{R}) \). In particular, \( (\mathcal{L}, \mathcal{R}) \) is not part of the input, but is instead replaced by a derivation oracle which answers, for any given \( S \subseteq \mathcal{L} \) and \( \pi \in \mathcal{L} \), whether \( \pi \) is among the conclusions of \( S \), i.e. \( \pi \in Cn(S) \). Such a derivation oracle operates in some complexity class \( D \) that depends on the deductive system \( (\mathcal{L}, \mathcal{R}) \). We say that \( D \) is the complexity class of the derivation problem for \( \mathcal{F} = (\mathcal{L}, \mathcal{R}, A, \neg) \).

Table 5 summarises the existing complexity upper bounds for ABA, side-by-side in cases of flat and generic (i.e. possibly non-flat) ABA frameworks.

<table>
<thead>
<tr>
<th>( sem )</th>
<th>( \text{VER} )</th>
<th>( \text{CA} )</th>
<th>( \text{SA} )</th>
<th>( \text{VER} )</th>
<th>( \text{CA} )</th>
<th>( \text{SA} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{adm} )</td>
<td>( \text{P}^D )</td>
<td>( \text{NP}^D )</td>
<td>trivial (D)</td>
<td>( \text{coNP}^D )</td>
<td>( \Sigma^P_2 )</td>
<td>( \Pi^P_2 )</td>
</tr>
<tr>
<td>( \text{prf} )</td>
<td>( \text{coNP}^D )</td>
<td>( \text{NP}^D )</td>
<td>( \Pi^P_2 )</td>
<td>( \text{coNP}^D )</td>
<td>( \Sigma^P_2 )</td>
<td>( \Pi^P_3 )</td>
</tr>
<tr>
<td>( \text{stb} )</td>
<td>( \text{P}^D )</td>
<td>( \text{NP}^D )</td>
<td>( \text{coNP}^D )</td>
<td>( \text{P}^D )</td>
<td>( \text{NP}^D )</td>
<td>( \text{coNP}^D )</td>
</tr>
</tbody>
</table>

Table 5: Existing upper bounds for complexity of ABA. \( D \) is the complexity class of the derivation problem.

The lower bounds however, i.e. the hardness results, depend on the concrete instance of ABA, in particular, on the exact complexity \( D \) of the derivation problem. One well-known instance of ABA studied in [32] is the flat Logic Programming ABA instance, called \( LP-ABA \) henceforth. An LP-ABA framework \( \mathcal{F} = (\mathcal{L}, \mathcal{R}, A, \neg) \) is defined thus. For a logic program \( P \) with Herbrand base \( \mathcal{H}B \) (consisting of ground atoms, which are atomic formulas all of whose arguments are ground terms, where ground terms are (recursively defined as) either constant symbols or function symbols (that appear in \( P \)) applied to ground terms) and \( \mathcal{H}B_{\text{not}} = \{ \neg a : a \in \mathcal{H}B \} \):

- \( \mathcal{L} = \mathcal{H}B \cup \mathcal{H}B_{\text{not}} \),
- \( \mathcal{R} = P \),
- \( A = \mathcal{H}B_{\text{not}} \),
- \( \neg a = a \quad \forall \neg a \in A \).

LP-ABA in particular has its derivation problem, and more generally the problem of finding the conclusions of a set of assumptions, in \( P \) (cf. e.g. [32, 39] and Lemma 4.22 in Section 4.2.1). Table 6 summarises the existing complexity results for LP-ABA.

<table>
<thead>
<tr>
<th>( sem )</th>
<th>( \text{VER} )</th>
<th>( \text{CA} )</th>
<th>( \text{SA} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{adm} )</td>
<td>( \text{P} )</td>
<td>( \text{NP-c} )</td>
<td>( \text{P} )</td>
</tr>
<tr>
<td>( \text{prf} )</td>
<td>( \text{coNP-c} )</td>
<td>( \text{NP-c} )</td>
<td>( \Pi^P_2 )</td>
</tr>
<tr>
<td>( \text{stb} )</td>
<td>( \text{P} )</td>
<td>( \text{NP-c} )</td>
<td>( \text{coNP-c} )</td>
</tr>
</tbody>
</table>

Table 6: Existing lower bounds for complexity in LP-ABA.
Tables 5 and 6 indicate which classes the problems belong to and/or are complete for: for instance, $\text{VER}_{\text{adm}}$ is in $\mathcal{P}^D$ for flat ABA, but in $\text{coNP}^D$ for generic ABA; $\text{CA}_{\text{stb}}$ is in $\mathcal{NP}^D$ for ABA (generic as well as flat frameworks); $\text{SA}_{\text{adm}}$ has trivial complexity in flat ABA, equivalent to that of the derivation problem, namely $\mathcal{P}$; $\text{SA}_{\text{prf}}$ is $\Pi_{2}^{P}$-c for LP-ABA.

3.3. Existence Problem

We identify that $\text{SA}$ may be seen to be generally ill-formulated with respect to, particularly, stable and set-stable semantics. Indeed, since (set-)stable or extensions need not exist in general, $\text{SA}$ may be undesirably trivialised in such cases, as exemplified next.

Example 3.3. Consider an ABA framework $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{O})$ with $\mathcal{A} = \{a\}$ and $\mathcal{R} = \{a \leftarrow a\}$. In $\mathcal{F}$, $\{a\}$ attacks itself, so is not conflict-free. The only conflict-free set is $\emptyset$. But as $\emptyset$ does not attack $\{a\}$, $\emptyset$ is not stable. Hence, $\mathcal{F}$ has no stable or set-stable extension. Thus, by definition of $\text{SA}$ as in Section 3.2, any sentence is sceptically accepted under stable or set-stable semantics in $\mathcal{F}$. This is undesirable.

One way to solve this issue would be to add to the sceptical acceptance problem a mandatory check for the existence of an extension for the semantics considered. However, that would generally result in additional complexity for this problem. Indeed, in that case, to determine whether a sentence is sceptically accepted under a given semantics in an ABA framework, one would first need to check if there exists an extension under the semantics considered and then check if the sentence is sceptically accepted. Overall, the complexity of such a joint problem would be at least the complexity of the most complicated problem between existence and standard sceptical acceptance problems. The complexity results for such a joint problem would thus not match the results existing in literature. Instead, we study the complexity of the existence problem separately and slightly modify the definition of the sceptical acceptance problem to assume existence, as follows.

Definition 3.1 (Existence). Let $sem \in \{\text{adm}, \text{prf}, \text{stb}, \text{set-stb}, \text{cpl}, \text{w-f}\}$ be given. The existence problem, denoted $\text{EX}_{sem}$, is the problem of deciding whether there exists a $sem$ extension $S \subseteq \mathcal{A}$ of $\mathcal{F}$.

Given the possible shortcomings of sceptical acceptance in the light of existence, we redefine the sceptical acceptance problem as follows.

Definition 3.2 (Sceptical Acceptance Revisited). Let $sem \in \{\text{adm}, \text{prf}, \text{stb}, \text{set-stb}, \text{cpl}, \text{w-f}\}$. The sceptical acceptance problem for a given $\pi \in \mathcal{L}$, denoted $\text{SA}_{sem}(\pi)$, is the problem of deciding whether for each $sem$ extension $S$ of $\mathcal{F}$ it holds that $\pi \in \text{Cn}(S)$, assuming that there exists at least one $sem$ extension of $\mathcal{F}$.

This rectification of $\text{SA}$ is a formalisation that preserves the intended meaning of sceptical acceptance in the absence of extensions under a given semantics. It also allows to preserve the complexity results established in the literature under the assumption that the existence check for extensions is not part of the problem.
3.4. Complexity Problems in ABA

The following are then the complexity problems of interest in ABA. For an ABA framework \( F \) and \( \text{sem} \in \{ \text{adm}, \text{prf}, \text{stb}, \text{set-stb}, \text{cpl}, \text{w-f}, \text{grd} \} \):

- The set verification problem for a given \( S \subseteq A \), denoted \( \text{VER}_{\text{sem}}(S) \), is the problem of deciding whether \( S \) is a \( \text{sem} \) extension of \( F \).
- The existence problem, denoted \( \text{EX}_{\text{sem}} \), is the problem of deciding whether there exists a \( \text{sem} \) extension \( S \subseteq A \) of \( F \).
- The credulous acceptance problem for a given \( \pi \in L \), denoted \( \text{CA}_{\text{sem}}(\pi) \), is the problem of deciding whether there exists a \( \text{sem} \) extension \( S \subseteq A \) such that \( \pi \in Cn(S) \).
- The sceptical acceptance problem for a given \( \pi \in L \), denoted \( \text{SA}_{\text{sem}}(\pi) \), is the problem of deciding whether for each \( \text{sem} \) extension \( S \subseteq A \) it holds that \( \pi \in Cn(S) \), assuming that there exists at least one \( \text{sem} \) extension of \( F \).

4. Complexity Results for ABA

We here establish new results regarding the complexity of ABA that will be necessary to establish the complexity of PABA. In particular, the complexity of the probabilistic counterparts of the problems of verification and acceptance in PABA (to be introduced later in Section 5.3) will depend on the complexity of verification and acceptance in ABA. This holds under each semantics considered in this paper, so we need to establish ABA complexity results under the semantics for which they are not known. Further, as stated in Section 3, the complexity class membership results can be established only with reference to the complexity of the derivability in the underlying logic of ABA frameworks, and the hardness results are ABA instance-specific. The same holds for the probabilistic complexity problems in PABA, so we will need ABA upper bounds with reference to the complexity class \( D \) of a derivation oracle and lower bounds with respect to the concrete instantiating logics and the exact \( D \). As flat and non-flat ABA frameworks are known to be of different complexity under most semantics, we consider problems in both Flat and Generic ABA. For instance-specific lower bounds, we chose the well-known instance LP-ABA in the flat case, and extend it naturally to a non-flat instance in the general case. We will use the complexity results pertaining to both Flat and Generic ABA as well as their instances to establish the corresponding probabilistic complexity results in PABA.

In this section we use many-one reductions (see Section 2.2).

4.1. Upper Bounds for ABA

We establish the upper bounds for verification (\( \text{VER} \)), existence (\( \text{EX} \)), credulous (\( \text{CA} \)) and sceptical (\( \text{SA} \)) acceptance problems in ABA where they do not already exist. We first complement the existing results (see Section 3.2) for admissible (\( \text{adm} \)), preferred (\( \text{prf} \)) and stable (\( \text{stb} \)) semantics with new results for the existence problem under these semantics. We then consider the remaining semantics of set-stable (\( \text{set-stb} \)), complete (\( \text{cpl} \)) and well-founded/grounded (\( \text{w-f} / \text{grd} \)) extensions for all four problems of interest. We establish results for both flat and generic (possibly non-flat) ABA.

In what follows, we assume the derivation problem for \( F = (L, \mathcal{R}, A, \cdot, \cdot) \) to be in an hitherto unspecified complexity class \( D \) when considering the complexity upper bounds for ABA (as discussed in Section 3.2).
4.1.1. Complementing the Existing Results for Admissible, Preferred and Stable Semantics

First, existence under admissible semantics is no harder than credulous acceptance:

**Theorem 4.1.** For a generic ABA framework $\mathcal{F}$, $\text{EX}_{adm}$ is in $(\Sigma^P_2)^D$.

*Proof.* $\text{EX}_{adm}$ can be solved by guessing a set of assumptions and checking if it is an admissible extension (cf. Section 2.1). An oracle that answers admissibility can be taken in coNP$^D$ (see Table 5 in Section 3.2). So $\text{EX}_{adm}$ is in $\text{NP}^{\text{coNP}^D} = \text{NP}^{\text{NP}^D} = (\Sigma^P_2)^D$. □

The same holds under preferred semantics, as the following two results show.

**Lemma 4.2.** For a generic ABA framework $\mathcal{F}$, $\text{EX}_{prf}$ and $\text{EX}_{adm}$ are equivalent.

*Proof.* If $\mathcal{F}$ has an admissible extension, then it also has a preferred extension, namely any $\subseteq$-maximally admissible one. Conversely, if $\mathcal{F}$ has a preferred extension, then it is admissible by definition, and so $\mathcal{F}$ has an admissible extension. Thus, $\mathcal{F}$ has an admissible extension iff $\mathcal{F}$ has a preferred extension. Consequently, $\text{EX}_{prf}$ and $\text{EX}_{adm}$ are equivalent. □

**Corollary 4.3.** For a generic ABA framework $\mathcal{F}$, $\text{EX}_{prf}$ is in $(\Sigma^P_2)^D$.

*Proof.* $\text{EX}_{adm}$ is in $(\Sigma^P_2)^D$ by Theorem 4.1, and $\text{EX}_{prf}$ is equivalent to $\text{EX}_{adm}$ by Lemma 4.2, so $\text{EX}_{prf}$ is in $(\Sigma^P_2)^D$. □

In flat ABA, on the other hand, existence under admissible and preferred semantics is trivial:

**Theorem 4.4.** For a flat ABA framework $\mathcal{F}$, $\text{EX}_{adm}$ and $\text{EX}_{prf}$ are trivial with answer ‘Yes’.

*Proof.* If $\mathcal{F}$ is flat, it necessarily admits a preferred (and hence admissible) extension, according to [40, Theorem 2.20]. So $\text{EX}_{adm}$ and $\text{EX}_{prf}$ are trivial. □

Existence of stable extensions is not guaranteed in either generic or flat ABA, and so the existence problem is non-trivial under stable semantics:

**Theorem 4.5.** For a generic ABA framework $\mathcal{F}$, $\text{EX}_{stb}$ is in $\text{NP}^D$.

*Proof.* Guess a set of assumptions and check if it is stable, using a $P^D$-oracle (see Table 5 in Section 3.2). This solution puts $\text{EX}_{stb}$ in $\text{NP}^D$. □

We now turn to the other semantics.
4.1.2. New Results for Set-stable Semantics

We start with set-stable semantics, which turns out to be on the same level as stable semantics. We first consider the fundamental verification problem.

**Theorem 4.6.** For a generic ABA framework $F$, $\text{VER}_{\text{set-stb}}$ is in $P^D$.

**Proof.** Let $S \subseteq A$. To determine if $S$ is set-stable, we use the following algorithm.

1. Check that $S$ is closed using $|A\setminus S|$ calls to a $D$-oracle, determining for each $a \in A \setminus S$ if $a \in Cn(S)$.
2. Check that $S$ is conflict-free using $|S|$ calls to a $D$-oracle, determining for each $a \in S$ if $a \in Cn(S)$.
3. Compute the set $A$ of assumptions not attacked by $S$ and not belonging to $S$ using $|A\setminus S|$ calls to $D$-oracle, determining for each $a \in A \setminus S$ if $a \not\in Cn(S)$.
4. For every $a \in A$, compute $Cn(\{a\}) \cap A$ using at most $|A|$ calls to a $D$-oracle, determining for each $b \in A$ if $b \in Cn(\{a\})$. If $Cn(\{a\}) \cap A \subseteq A \cup S$, return ‘No’. (Indeed, if the closure $Cn(\{a\})$ of at least one assumption $a \in A \setminus S$ is not attacked by $S$, then $S$ is not set-stable.)

The algorithm shows that $\text{VER}_{\text{set-stb}}$ is in $P^D$. \hfill $\blacksquare$

Given the complexity of verification, we can establish the complexity of the other problems under set-stable semantics.

**Theorem 4.7.** For a generic ABA framework $F$, $\text{EX}_{\text{set-stb}}$ is in $NP^D$, $\text{CA}_{\text{set-stb}}$ is in $NP^D$, and $\text{SA}_{\text{set-stb}}$ is in $coNP^D$.

**Proof.** To determine existence, guess a set of assumptions and check if it is set-stable. Since $\text{VER}_{\text{set-stb}}$ is in $P^D$ by Theorem 4.6, $\text{EX}_{\text{set-stb}}$ is in $NP^D$.

For credulous acceptance, to determine whether there is a set-stable extension whose conclusions include a given sentence $\pi \in L$, we can additionally use a $D$-oracle, which puts $\text{CA}_{\text{set-stb}}$ in $NP^D$ too.

For sceptical acceptance, to solve its complement problem of deciding whether $\pi \in L$ is not among the conclusions of every set-stable extension, guess $S \subseteq A$, check that it is set-stable, and check $\pi \not\in Cn(S)$. The latter problem is in $NP^D$, so $\text{SA}_{\text{set-stb}}$ is in $coNP^D$. \hfill $\blacksquare$

We can now also observe that the same upper bounds hold in flat ABA under set-stable semantics, because in flat ABA, set-stable semantics coincides with stable semantics (as stated in [36, Proposition 3], proof of which is not present in the paper in question).

**Corollary 4.8.** For a flat ABA framework $F$, $\text{VER}_{\text{set-stb}}$, $\text{EX}_{\text{set-stb}}$, and $\text{CA}_{\text{set-stb}}$ are in $NP^D$, and $\text{SA}_{\text{set-stb}}$ is in $coNP^D$.

**Proof.** If $F$ is flat, all sets of assumptions are closed by definition. If $S \subseteq A$ is set-stable, it is closed, conflict-free and attacks $Cn(\{a\}) \cap A = \{a\}$ for every $a \in A \setminus S$, and so is stable. Likewise, if $S \subseteq A$ is stable, it is closed, conflict-free and attacks $\{a\} = Cn(\{a\}) \cap A$ for every $a \in A \setminus S$, and so is set-stable. Consequently, $\text{VER}_{\text{set-stb}}$, $\text{EX}_{\text{set-stb}}$, $\text{CA}_{\text{set-stb}}$ and $\text{SA}_{\text{set-stb}}$ are equivalent to their counterparts under stable semantics for flat ABA, whence their complexity upper bounds follow from Theorems 4.6 and 4.7. \hfill $\blacksquare$
4.1.3. New Results for Complete Semantics

Since complete extensions are admissible and additionally contain all assumptions they defend, we expect reasoning under complete semantics to be slightly harder than under admissible semantics. We show that this is indeed the case for the verification problem.

**Theorem 4.9.** For a generic ABA framework $\mathcal{F}$, $\text{VER}_{cpl}$ is in $\text{DP}^\text{D}$.

**Proof.** Let $S \subseteq A$. To determine if $S$ is complete, we need to check if it is admissible and also if $\text{Def}(S) \subseteq S$ (see Section 3.1). The admissibility check can be done by a coNP$^\text{D}$-oracle (see Table 5 in Section 3.2). Now, determining $\text{Def}(S) \subseteq S$ is equivalent to determining that every $a \in A \setminus S$ is not defended by $S$. The latter problem can be solved as follows. For every $a \in A \setminus S$, guess $A \subseteq A$ and then:

1. Check that $A$ is closed using $|A \setminus A|$ calls to a D-oracle.
2. Check that $A$ attacks $a$ using a single call to a D-oracle.
3. Check that $A$ is not attacked by $S$ using $|A|$ calls to a D-oracle.

This algorithm puts the above problem, and consequently the problem of determining $\text{Def}(S) \subseteq S$, in NP$^\text{D}$. All in all, to solve $\text{VER}_{cpl}$ we can solve a problem in each of coNP$^\text{D}$ and NP$^\text{D}$, which means $\text{VER}_{cpl}$ is in DP$^\text{D}$. $\square$

Verification under complete semantics in flat ABA is easier:

**Theorem 4.10.** For a flat ABA framework $\mathcal{F}$, $\text{VER}_{cpl}$ is in $\text{P}^\text{D}$.

**Proof.** Let $S \subseteq A$. To determine if $S$ is complete, we use the following algorithm.

1. Compute the set $A$ of assumptions attacked by $S$ using $|A|$ calls to a D-oracle.
2. Check that $S$ is conflict-free by checking that $A \cap S = \emptyset$.
3. Check that $S$ is admissible by checking that it is not attacked by $A \setminus A$ using $|S|$ calls to a D-oracle.
4. Check that $S$ is complete by checking that it does not defend any $a \in A \setminus S$, which can be done by checking that $A \setminus A$ attacks every $a \in A \setminus S$ using $|A \setminus S|$ calls to a D-oracle.

This algorithm ensures $\text{VER}_{cpl}$ is in P$^\text{D}$. $\square$

We next use the line of reasoning as in [32, (proof of) Theorem 8] for credulous acceptability under admissible semantics to establish the complexity of credulous acceptability under complete semantics:

**Theorem 4.11.** For a generic ABA framework $\mathcal{F}$, $\text{CA}_{cpl}$ is in $(\Sigma^\text{P}_2)^\text{D}$.

**Proof.** Guess a set $S$ of assumptions and check if it is complete, using a DP$^\text{D}$-oracle (see Theorem 4.9). If $S$ is complete, check if $\pi \in \text{Cn}(S)$ using a call to D. This proves $\text{CA}_{cpl}$ in NP$^{\text{DP}^\text{D}} = (\Sigma^\text{P}_2)^\text{D}$. $\square$

We can now prove the following results.

**Theorem 4.12.** For a generic ABA framework $\mathcal{F}$, $\text{EX}_{cpl}$ is in $(\Sigma^\text{P}_2)^\text{D}$.
Proof. Guess a set of assumptions and check if it is complete, using a D^P^D-oracle.

**Corollary 4.13.** For a flat ABA framework \( \mathcal{F} \), \( \text{Ex}_{\text{cpl}} \) is trivial with answer ‘Yes’.

Proof. According to [40, Theorem 2.20], \( \mathcal{F} \) admits a complete extension.

**Lemma 4.14.** For a flat ABA framework \( \mathcal{F} \), \( \text{Ck}_{\text{cpl}} \) and \( \text{Ck}_{\text{adm}} \) are equivalent.

Proof. Complete extensions are admissible by definition, so being credulously accepted under complete semantics implies being credulously accepted under admissible semantics. Conversely, every admissible extension admits a preferred extension as a superset [40, Theorem 2.11(ii)]. Further, in a flat \( \mathcal{F} \), preferred extensions are complete [40, Theorem 2.14(i)]. Thus, being credulously accepted under admissible semantics implies being credulously accepted under complete semantics. Consequently, \( \text{Ck}_{\text{cpl}} \) and \( \text{Ck}_{\text{adm}} \) are equivalent.

**Corollary 4.15.** For a flat ABA framework \( \mathcal{F} \), \( \text{Ck}_{\text{cpl}} \) is in \( \text{NP}^D \).

Proof. Since \( \text{Ck}_{\text{adm}} \) is in \( \text{NP}^D \) for flat ABA (Table 5 in Section 3.2), \( \text{Ck}_{\text{cpl}} \) in \( \text{NP}^D \) for flat ABA follows from Lemma 4.14.

We finish dealing with complete semantics by considering the sceptical acceptance problem, which, specifically in the flat case, is harder than under admissible semantics though not harder than under stable/set-stable semantics.

**Theorem 4.16.** For a generic ABA framework \( \mathcal{F} \), \( \text{Sa}_{\text{cpl}} \) is in \( (\Sigma^P_2)^D \).

Proof. Consider the complement problem of \( \text{Sa}_{\text{cpl}} \), namely determining whether there is a complete extension \( S \) of \( \mathcal{F} \) such that \( \pi \notin \text{Cn}(S) \). This problem can be solved by guessing \( S \), checking that it is complete and that \( \pi \notin \text{Cn}(S) \) using a D-oracle. As \( \text{Ver}_{\text{cpl}} \) is in \( \text{DP}^D \) by Theorem 4.9, this problem is in \( \text{NP}^{\text{DP}^D} = (\Sigma^P_2)^D \). Hence, \( \text{Sa}_{\text{cpl}} \) is in \( \text{coNP}^{\text{DP}^D} = (\Pi^P_2)^D \).

Finally, sceptical acceptance is considerably easier under complete semantics in flat ABA. The following result will follow from the results on the acceptance under grounded semantics, which we derive in the next section.

**Corollary 4.17.** For a flat ABA framework \( \mathcal{F} \), \( \text{Sh}_{\text{cpl}} \) is in \( P^D \).

Proof. In Section 4.1.4.

4.1.4. New Results for Well-founded/Grounded Semantics

We finish our exposition of new upper bounds for ABA with results concerning well-founded (grounded in the flat case) semantics. We first observe that as the well-founded extension is defined as the intersection of complete extensions, its existence coincides with the existence of complete extensions:

**Corollary 4.18.** For a generic ABA framework \( \mathcal{F} \), \( \text{Ex}_{w-f} \) is in \( (\Sigma^P_2)^D \).

Proof. \( \text{Ex}_{w-f} \) is equivalent to \( \text{Ex}_{\text{cpl}} \) by definition of well-founded semantics, so \( \text{Ex}_{w-f} \) is in \( (\Sigma^P_2)^D \) just as \( \text{Ex}_{\text{cpl}} \) is, according to Theorem 4.12.
In terms of verification, we expect and indeed prove next that it is harder than under other semantics considered in this paper.

**Theorem 4.19.** For a generic ABA framework $\mathcal{F}$, $\text{VER}_{w-f}$ is in $(\Delta^p_3)^D$.

**Proof.** We first show that verifying that any one assumption does not belong to the well-founded extension (assuming it exists) is in $(\Sigma^P_2)^D$. Let $a \in A$. Consider the following algorithm.

1. Guess $S \subseteq A$.
2. Check that $S$ is complete using a $\text{DP}^D$-oracle.
3. Check that $a \not\in S$.

This algorithm puts the above problem in $\text{NP}^{\text{DP}^D} = (\Sigma^P_2)^D$. Suppose we have an oracle $O$ that solves the above problem, as well as its complement of verifying if $a \in A$ belongs to the well-founded extension (assuming existence). We can thus use the following algorithm to verify if a given $S \subseteq A$ is well-founded.

1. Check if the well-founded extension exists using a $(\Sigma^P_2)^D$-oracle (Corollary 4.18).
2. For every $a \in S$, check that it belongs to the well-founded extension using $|S|$ calls to $O$.
3. For every $a \in A \setminus S$, check that it does not belong to the well-founded extension using $|A \setminus S|$ calls to $O$.

This algorithm ensures $\text{VER}_{w-f}$ is in $\text{P}^{(\Sigma^P_2)^D} = (\Delta^p_3)^D$.

Credulous and sceptical acceptance under well-founded semantics fall into the same class as verification, because a $(\Delta^p_3)^D$-oracle can be used to compute the well-founded extension:

**Corollary 4.20.** For a generic ABA framework $\mathcal{F}$, $\text{CA}_{w-f}$ and $\text{SA}_{w-f}$ are in $(\Delta^p_3)^D$.

**Proof.** Since the well-founded extension, if it exists, is unique, $\text{CA}_{w-f}$ and $\text{SA}_{w-f}$ coincide. Further, the proof of Theorem 4.19 shows that the well-founded extension can be constructed (by exhaustively checking for every $a \in A$) using a $(\Delta^p_3)^D$-oracle. Checking if $\pi \in L$ belongs to the conclusions of the well-founded extension then simply requires a D-oracle call. Hence, $\text{CA}_{w-f}$ and $\text{SA}_{w-f}$ are in $(\Delta^p_3)^D$.

In flat ABA, however, the upper bounds for grounded semantics are all in $\text{P}^D$.

**Theorem 4.21.** For a flat ABA framework $\mathcal{F}$, $\text{VER}_{grd}$, $\text{CA}_{grd}$ and $\text{SA}_{grd}$ are in $\text{P}^D$, and $\text{EX}_{grd}$ is trivial with answer ‘Yes’.

**Proof.** If $\mathcal{F}$ is flat, then according to [40, Theorem 2.20], the grounded extension always exists. Further, computing the grounded extension can be done in polynomial time with a D-oracle, due to its characterisation as the least fix point of the defence operator $\text{Def}$ (see Section 3.1). Verification is therefore no harder, and hence in $\text{P}^D$. Since the grounded extension is unique, $\text{CA}_{grd}$ and $\text{SA}_{grd}$ coincide, and are in $\text{P}^D$ too.

We finally prove that the upper bound of sceptical acceptance under complete semantics in flat ABA is in $\text{P}^D$, given the results on acceptance under grounded semantics.
Proof of Corollary 4.17. If \( \pi \in \mathcal{L} \) is sceptically accepted under complete semantics, then \( \pi \in \text{Cn}(\mathcal{G}) \) for the grounded extension \( \mathcal{G} \), which is complete according to [29, Theorem 6.2]. Conversely, if \( \pi \in \text{Cn}(\mathcal{G}) \), then \( \pi \) is sceptically accepted under complete semantics, by definition of the grounded extension \( \mathcal{G} \). So \( \mathcal{SA}_{cpl} \) is equivalent to \( \mathcal{CA}_{grd} \). The latter is in \( \mathcal{P}^D \), by Theorem 4.21. So \( \mathcal{SA}_{cpl} \) is in \( \mathcal{P}^D \) too.

Table 7 (same as Table 2 in Section 1 but reiterated here for convenience) summarises the existing (given in Section 3.2) and new (this section, Section 4.1) results on the complexity upper bounds of ABA.

<table>
<thead>
<tr>
<th>( \text{sem} )</th>
<th>( \text{VER} )</th>
<th>( \text{EX} )</th>
<th>( \text{CA} )</th>
<th>( \text{SA} )</th>
<th>( \text{VER} )</th>
<th>( \text{EX} )</th>
<th>( \text{CA} )</th>
<th>( \text{SA} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>adm</td>
<td>( \mathcal{P}^D )</td>
<td>trivial (Yes)</td>
<td>( \mathcal{NP}^D )</td>
<td>D</td>
<td>co( \mathcal{NP}^D )</td>
<td>( \sum_2^D )</td>
<td>( \Delta_2^D )</td>
<td>( \Pi_2^D )</td>
</tr>
<tr>
<td>cpl</td>
<td>( \mathcal{P}^D )</td>
<td>trivial (Yes)</td>
<td>( \mathcal{NP}^D )</td>
<td>( \mathcal{P}^D )</td>
<td>DP( \mathcal{D} )</td>
<td>( \sum_2^D )</td>
<td>( \Delta_2^D )</td>
<td>( \Pi_2^D )</td>
</tr>
<tr>
<td>prf</td>
<td>co( \mathcal{NP}^D )</td>
<td>trivial (Yes)</td>
<td>( \mathcal{NP}^D )</td>
<td>( \Pi_2^D )</td>
<td>( \mathcal{NP}^D )</td>
<td>( \mathcal{NP}^D )</td>
<td>( \mathcal{NP}^D )</td>
<td>( \mathcal{NP}^D )</td>
</tr>
<tr>
<td>stb</td>
<td>( \mathcal{P}^D )</td>
<td>( \mathcal{NP}^D )</td>
<td>( \mathcal{NP}^D )</td>
<td>co( \mathcal{NP}^D )</td>
<td>( \mathcal{P}^D )</td>
<td>( \mathcal{NP}^D )</td>
<td>( \mathcal{NP}^D )</td>
<td>co( \mathcal{NP}^D )</td>
</tr>
<tr>
<td>set-stb</td>
<td>( \mathcal{P}^D )</td>
<td>( \mathcal{NP}^D )</td>
<td>( \mathcal{NP}^D )</td>
<td>co( \mathcal{NP}^D )</td>
<td>( \mathcal{P}^D )</td>
<td>( \mathcal{NP}^D )</td>
<td>( \mathcal{NP}^D )</td>
<td>co( \mathcal{NP}^D )</td>
</tr>
<tr>
<td>grd</td>
<td>( \mathcal{P}^D )</td>
<td>trivial (Yes)</td>
<td>( \mathcal{NP}^D )</td>
<td>( \mathcal{P}^D )</td>
<td>( \mathcal{NP}^D )</td>
<td>( \mathcal{NP}^D )</td>
<td>( \mathcal{NP}^D )</td>
<td>( \mathcal{NP}^D )</td>
</tr>
<tr>
<td>w-f</td>
<td>( \mathcal{P}^D )</td>
<td>trivial (Yes)</td>
<td>( \mathcal{NP}^D )</td>
<td>( \mathcal{P}^D )</td>
<td>( \mathcal{NP}^D )</td>
<td>( \mathcal{NP}^D )</td>
<td>( \mathcal{NP}^D )</td>
<td>( \mathcal{NP}^D )</td>
</tr>
</tbody>
</table>

Table 7: Summary of old and new (in bold) upper bounds for classical complexity problems in ABA. \( \mathcal{D} \) is the complexity class of the derivation problem.

4.2. Lower Bounds for ABA

Lower bounds can only be established for ABA frameworks instantiated with an underlying deductive system. We consider propositional instances of ABA. Specifically, in Section 4.2.1 we study arguably the most widely known instance of flat ABA, namely LP-ABA, with some existing complexity results established in [32] (as overviewed in Section 3.2). We complement the existing complexity lower bounds with new results under complete and grounded semantics, as well as with the existence problem under all semantics in \{adm, prf, stb, cpl, grd\}. In Section 4.2.2 we define another instance of ABA, called Horn-ABA, that is potentially non-flat and subsumes LP-ABA. We then study its complexity lower bounds and establish results for all the four problems of verification, existence, credulous and sceptical acceptance under all semantics in \{adm, prf, stb, set-stb, cpl, w-f\}.

4.2.1. Complexity of LP-ABA

Before we delve into the complexity problems of interest as stated in Section 3.4, we give a general result that in LP-ABA, finding the conclusions of a given set of assumptions is a problem in \( \mathcal{P} \). By subsumption, this implies that the derivation problem of determining whether a given sentence is a conclusion of a given set of assumptions (see Section 3.2) is in \( \mathcal{P} \). The latter echoes the statements in [39] that a derivation oracle for (flat) LP-ABA operates in \( \mathcal{P} \) as well as in [32] that the derivation problem in propositional Horn logic is in \( \mathcal{P} \); we nonetheless give a brief proof of the result for the sake of completeness.

Lemma 4.22. For a flat LP-ABA framework \( \mathcal{F} \) and \( S \subseteq A \), computing the conclusions \( \text{Cn}(S) = \{ \phi \in \mathcal{L} : \exists A \subseteq S, A \vdash \phi \} \) is in \( \mathcal{P} \).
Proof. Let \( T(S) = S \cup \{ \varphi \in L : \exists r \in R, \text{head}(r) = \varphi, \text{body}(r) \subseteq S \} \) (see Section 3.1). Then \( Cn(S) = T^{\|R\|}(S) \). Since a logic program is always finite, \(|R|\) is finite. Thus, computing \( Cn(S) \) is polynomial in time.

This result particularly implies that in LP-ABA, a derivation oracle can be taken to operate in \( D = P \). It also entails that some problems in LP-ABA are easy.

**Corollary 4.23.** For a flat LP-ABA framework \( F \), \( \text{VER}_{\text{grad}}, \text{CA}_{\text{grad}} \) and \( \text{SA}_{\text{grad}} \) are in \( P \).

**Proof.** For flat ABA in general, \( \text{VER}_{\text{grad}}, \text{CA}_{\text{grad}} \) and \( \text{SA}_{\text{grad}} \) are in \( P^D \) with a \( D \)-derivation oracle, according to Theorem 4.21. Since \( D = P \) for LP-ABA by Lemma 4.22, these problems are in \( P \).

**Corollary 4.24.** For a flat LP-ABA framework \( F \), \( \text{VER}_{\text{cpl}} \) and \( \text{SA}_{\text{cpl}} \) are in \( P \).

**Proof.** For flat ABA in general, \( \text{VER}_{\text{cpl}} \) and \( \text{SA}_{\text{cpl}} \) are in \( P^D \), according to Theorem 4.10 and Corollary 4.17, respectively. From Lemma 4.22 it then follows that these problems as in \( P \) for LP-ABA.

As with upper bounds under complete semantics, the situation is analogous with lower bounds: credulous acceptance under complete semantics is harder than either verification or sceptical acceptance.

**Corollary 4.25.** For a flat LP-ABA framework \( F \), \( \text{CA}_{\text{cpl}} \) is \( \text{NP-c} \).

**Proof.** \( \text{CA}_{\text{cpl}} \) is equivalent to \( \text{CA}_{\text{adm}} \), by Lemma 4.14. Since \( \text{CA}_{\text{adm}} \) is \( \text{NP-c} \) for LP-ABA (Table 6 in Section 3.2), \( \text{CA}_{\text{cpl}} \) is \( \text{NP-c} \) too.

Regarding the existence problem, it is trivial under admissible, preferred, complete and grounded semantics:

**Corollary 4.26.** For a flat LP-ABA framework \( F \), \( \text{EX}_{\text{adm}}, \text{EX}_{\text{prf}}, \text{EX}_{\text{cpl}} \) and \( \text{EX}_{\text{grad}} \) are trivial with answer ‘Yes’.

**Proof.** Follows from Theorem 4.4, Corollary 4.13 and Theorem 4.21.

The more interesting case is existence under stable semantics.

**Theorem 4.27.** For a flat LP-ABA framework \( F \), \( \text{EX}_{\text{stb}} \) is \( \text{NP-c} \).

**Proof.** \( \text{EX}_{\text{stb}} \) is in NP since the derivation problem is in \( P \) for LP-ABA. To prove \( \text{NP-hardness} \), we reduce \( \text{CA}_{\text{stb}} \) to \( \text{EX}_{\text{stb}} \), where \( \text{CA}_{\text{stb}} \) is \( \text{NP-c} \) (Table 6 in Section 3.2).

For \( \pi \in L \), define \( f(F, \pi) = (\mathcal{L}', \mathcal{R}', \mathcal{A}', \mathcal{R}) =: F' \) thus:

\begin{itemize}
  \item \( \mathcal{A}' = \mathcal{A} \cup \{ \text{not } b \} \) with \( b \notin \mathcal{L} \);
  \item \( \text{not } b' = b \) and \( \text{not } a' = \text{not } a = a \) for \( \text{not } a \in \mathcal{A} \);
  \item \( \mathcal{R}' = \mathcal{R} \cup \{ b \leftarrow \pi, b \leftarrow \text{not } b \} \);
  \item \( \mathcal{L}' = \mathcal{L} \cup \{ b, \text{not } b \} \).
\end{itemize}
We next show that $\text{CA}_{stb}$ for $\mathcal{F}$ can be reduced to the existence problem under stable semantics by showing that if one can solve the latter problem in general, and consequently for the particular LP-ABA framework $\mathcal{F}'$, then one can solve the former problem for the given $\mathcal{F}$ and $\pi$.

If $S$ is stable in $\mathcal{F}$ and $\pi \in Cn(S)$, then $S$ attacks the self-attacking $\{\text{not } b\}$ in $\mathcal{F}'$, and so $S$ is clearly stable in $\mathcal{F}'$. Thus, if $\pi$ is credulously accepted under stable semantics in $\mathcal{F}$, then there exists a stable extension of $\mathcal{F}'$.

Reciprocally, if there exists a stable extension $S'$ of $\mathcal{F}'$, then $\{\text{not } b\}$ attacks itself. But then $S'$ is also stable in $\mathcal{F}$. $S'$ also has to attack $\{\text{not } b\}$ in $\mathcal{F}'$, which means $b \in Cn(S')$, and hence $\pi \in Cn(S')$ too. Thus, if there exists a stable extension of $\mathcal{F}'$, then $\pi$ is credulously accepted under stable semantics in $\mathcal{F}$.

It is plain to see that $f$ is computable in polynomial time, so that $\text{CA}_{stb}$ is reducible to $\text{EX}_{stb}$, whence $\text{EX}_{stb}$ is indeed $\text{NP}$-c.

Table 8 (part of Table 3 in Section 1) summarises the existing (given in Section 3.2) and new (this section, Section 4.2.1) results on the complexity of flat LP-ABA.

<table>
<thead>
<tr>
<th>sem</th>
<th>VER</th>
<th>EX</th>
<th>CA</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>adm</td>
<td>$\mathcal{P}$</td>
<td>trivial (Yes)</td>
<td>$\text{NP}$-c</td>
<td>$\mathcal{P}$</td>
</tr>
<tr>
<td>cpl</td>
<td>$\mathcal{P}$</td>
<td>trivial (Yes)</td>
<td>$\text{NP}$-c</td>
<td>$\mathcal{P}$</td>
</tr>
<tr>
<td>prf</td>
<td>$\text{coNP}$-c</td>
<td>trivial (Yes)</td>
<td>$\text{NP}$-c</td>
<td>$\text{NP}$-c</td>
</tr>
<tr>
<td>stb</td>
<td>$\mathcal{P}$</td>
<td>$\text{NP}$-c</td>
<td>$\text{NP}$-c</td>
<td>$\text{coNP}$-c</td>
</tr>
<tr>
<td>grd</td>
<td>$\mathcal{P}$</td>
<td>trivial (Yes)</td>
<td>$\mathcal{P}$</td>
<td>$\mathcal{P}$</td>
</tr>
</tbody>
</table>

Table 8: Summary of old and new (in bold) lower bounds for classical complexity problems in LP-ABA.

Overall, we see that reasoning under complete semantics is equally as hard as under admissible semantics, and also that reasoning under grounded semantics is easy (polynomial time). It is also worth noting that checking for existence of stable extensions in LP-ABA frameworks is non-tractable (nondeterministic polynomial time).

We now turn our attention to a novel generic instance of ABA.

4.2.2. Complexity of Horn-ABA

We define a novel ABA instance based on propositional Horn logic as the underlying logic and effectively extending LP-ABA to the generic, possibly non-flat setting. We first define this generic ABA instance which we call Horn-ABA and discuss its derivation problem. We then study the complexity of Horn-ABA.

Definition 4.1. A Horn-ABA framework $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, ^\sim)$ is defined thus:

- $(\mathcal{L}, \mathcal{R})$ consists of a set $\mathcal{R}$ of rules either of the form $b_0 \leftarrow b_1, \ldots, b_m$ for $m \geq 1$ or of the form $b_0$, for $b_i \in \mathcal{L}$ (for all $i \in \{0, \ldots, m\}$), and $\mathcal{L}$ is a set of ground atoms;
- $\mathcal{A} \subseteq \mathcal{L}$ is a set of ground atoms in $\mathcal{L}$.

21
We refer to such a framework as ‘Horn-ABA’ since \((L, R)\) is basically propositional Horn logic without integrity constraints (of the form \(\bot \leftarrow b_1, \ldots, b_m\), with \(\bot\) representing inconsistency). Since assumptions can unrestrictedly appear as either the heads of rules or the contraries of other assumptions, Horn-ABA frameworks can be non-flat.

Note well that Horn-ABA subsumes LP-ABA in the sense that any LP-ABA framework can be seen as a Horn-ABA framework. Indeed, intuitively, given an LP-ABA framework, the negation as failure literals of the form \(\neg a\) can be replaced by new atoms in a Horn-ABA framework (with an appropriately extended language) while preserving semantic correspondence – see [49] and [29, Section 2.2]. We will utilise this fact in some of the proofs to come.

Example 4.1. The flat ABA framework \(F\) from Example 3.1 is a Horn-ABA framework where \(L = \{A, A', E, E', no\_compl, no\_compl\}\) is comprised of ground atoms (constants). With the additional assumption \(no\_exacerb\) and rule \(no\_compl \leftarrow no\_exacerb\) (expressing that the belief in the absence of complications can be also based on the belief in the absence of exacerbations) it would become a non-flat Horn-ABA framework.

Note also that, effectively, a class of necessarily flat Horn-ABA frameworks was implicitly studied in [50] where Caminada and Schulz established semantic correspondence between flat Horn-ABA frameworks (in the terminology of this paper) and normal logic programs (logic programs without strong (explicit) negation and without disjunction in the heads of rules). In this same vein, Horn-ABA can be seen as an instance of ABA based on logic programs with negation as failure in the heads as studied by Inoue and Sakama in [51]. The study of formal relationships between semantics of Horn-ABA and the corresponding logic programming instance is beyond the scope of this paper and is left for future work (but see a brief discussion in Section 7).

Another known non-flat ABA instance is based on Autoepistemic Logic (AEL) [45] and was studied in [29, 32]. It is interesting that ABA instantiated with AEL attains the complexity lower bounds that match the upper bounds of generic ABA (Table 7 in Section 4.1), as discussed in [39]. This in particular shows that reasoning in general ABA cannot be easier than the established upper bounds indicate, at least under admissible, preferred and stable semantics. In this section we will establish new instantiated non-flat ABA complexity lower bounds under all semantics in \(\{adm, prf, stb, set-stb, cpl, w-f\}\). We will particularly show that Horn-ABA also attains the complexity lower bounds that match the upper bounds of generic ABA, even though reasoning in the underlying logic of Horn-ABA is easier than in AEL. So we will have found an instance of generic ABA which shows that reasoning in general ABA cannot be easier than the established upper bounds indicate, even though the derivation problem for Horn-ABA is easier than that for AEL-based ABA (where it is \(\text{coNP-c}\)), as our first result on the derivation problem for Horn-ABA shows:

Corollary 4.28. For a Horn-ABA framework \(F\) and \(S \subseteq A\), computing the conclusions \(Cn(S) = \{\phi \in L : \exists A \subseteq S, A \vdash \phi\}\) is in \(P\).

Proof. Verbatim to the proof of Lemma 4.22. \(\square\)
Complexity results for Horn-ABA. In what follows, we will often reduce problems of known complexity, such as satisfiability problems of formulas in conjunctive normal form of propositional clauses, to complexity problems in Horn-ABA. Therefore, given a collection of boolean variables and propositional formulas constructed using standard connectives, we cast them as primitive sentences in the language $L$ of Horn logic. The elements of the body and the head of a rule from $\mathcal{R}$ will therefore take the form of an arbitrary propositional formula, and assumptions will often take the form of boolean variables. (We will see the first example of this in the proof of Theorem 4.29 below.) Also, for $X$ a set of positive boolean variables, we use $\neg X$ to denote $\{\neg x : x \in X\}$.

We start with verification under complete semantics.

**Theorem 4.29.** For a Horn-ABA framework $\mathcal{F}$, VER$_{cpl}$ is DP-$c$.

**Proof.** First VER$_{cpl}$ is in DP because verification under complete semantics is in DP$^D$ for generic ABA (Table 7 in Section 4.1) and the derivation problem for $\mathcal{F}$ is in $D = P$, by Corollary 4.28.

To prove DP-hardness of VER$_{cpl}$, we reduce a DP-hard problem to it. Specifically, we consider 3-SAT-UNSAT, which is the decision problem taking in input two so-called 3CNF formulas $\Phi_1$ and $\Phi_2$ in conjunctive normal form (CNF) with 3 literals in each clause, and determining whether $\Phi_1$ is satisfiable and $\Phi_2$ is unsatisfiable. To this end, we construct a function $f$, which takes as input two 3CNF formulas and outputs an ABA framework. We start by defining a function $g$ taking as input one formula and we later use $g$ to define $f$.

A 3CNF formula $\Phi$ is of the form $\Phi = C^\Phi_1 \land \ldots \land C^\Phi_k$, where for all $1 \leq i \leq k$, $C^\Phi_i = x \lor y \lor z$ with $x, y, z \in X^\Phi$, where $X^\Phi$ the set of the positive boolean variables in $\Phi$. Define $g(\Phi) = \langle \mathcal{L}_\Phi, \mathcal{R}_\Phi, A_\Phi, \neg \rangle$ thus:

- $A_\Phi = X^\Phi \cup \neg X^\Phi \cup \{\text{contra}^\Phi, \text{logic}^\Phi, \text{unsat}^\Phi\}$;
- $\forall x \in X^\Phi \ x = \neg x, \neg \neg x = x$, $\text{contra}^\Phi = \text{logic}^\Phi$, $\text{unsat}^\Phi = \text{sat}^\Phi$;
- $\mathcal{R}_\Phi = \{C^\Phi_i \leftarrow x : x \in \{a, b, c\}, C^\Phi_i = a \lor b \lor c \cup \{\text{sat}^\Phi \leftarrow C^\Phi_1, \ldots, C^\Phi_k\}$
  $\cup \{\text{contra}^\Phi \leftarrow x, \neg x : x \in X^\Phi\}$.

Intuitively, $g(\Phi)$ encodes the formula $\Phi$ in a non-flat ABA framework. A truth assignment for the variables in $X$ that satisfies $\Phi$ will have $\text{sat}^\Phi$ among its conclusions. However, by choosing a set of assumptions from $A$ we could make inconsistent choices, such as choosing $x$ and $\neg x$ for some $x \in X^\Phi$. To account for this, assumption $\text{contra}^\Phi$ is introduced. It also results into non-flatness of the framework.

We now express these ideas formally for our reduction purposes. First, we prove that $\Phi$ is unsatisfiable iff $\{\text{unsat}^\Phi, \text{logic}^\Phi\}$ is an admissible extension of $g(\Phi)$.

- Suppose $\Phi$ is satisfiable. Then there exists a truth assignment $X \subseteq X^\Phi \cup \neg X^\Phi$ satisfying $\Phi$. $X$ is a truth assignment, so there is no $x \in X$ such that $\neg x \in X$. Thus, $\text{contra}^\Phi \notin \text{Cn}(X)$. As $\text{contra}^\Phi$ is the only assumption in the head of a rule, $X$ is closed. Additionally, $X$ satisfies $\Phi$, so $\text{sat}^\Phi \in \text{Cn}(X)$, whence $X$ attacks $\text{unsat}^\Phi$. As neither $\{\text{unsat}^\Phi\}$ nor $\{\text{logic}^\Phi\}$ attacks $X$, and since $X$ is closed, $\{\text{unsat}^\Phi, \text{logic}^\Phi\}$ does not defend against $X$, whence $\{\text{unsat}^\Phi, \text{logic}^\Phi\}$ is not admissible.
• Suppose \( \Phi \) is unsatisfiable. Since \( Cn(\{unsat^\Phi,logic^\Phi\}) = \{unsat^\Phi,logic^\Phi\} \), we have that \( \{unsat^\Phi,logic^\Phi\} \) is closed and conflict-free. We show that it additionally defends itself. So let \( S \subseteq A \) be closed and attacking \( \{unsat^\Phi,logic^\Phi\} \). Since \( logic^\Phi \) cannot be attacked, \( S \) attacks \( unsat^\Phi \), and so \( A \vdash sat^\Phi \) for some \( A \subseteq S \). This can happen only if \( A \vdash C^\Phi \) for every \( i \in \{1,\ldots,k\} \). Let \( X' \) be a \( \subseteq \)-maximal subset of \( X^\Phi \cup \neg X^\Phi \) contained in \( A \). \( X' \) is non-empty since \( A \vdash C^\Phi \) for every \( i \in \{1,\ldots,k\} \). Then \( A = X' \cup B \) with \( B \subseteq \{contra^\Phi,unsat^\Phi,logic^\Phi\} \). Thus, by construction of \( g(\Phi) \), \( X' \vdash sat^\Phi \). So, if \( X' \) were a partial truth assignment for the variables in \( X^\Phi \), then any truth assignment extending \( X' \) would be a satisfying truth assignment for \( \Phi \). But \( \Phi \) is unsatisfiable, so \( X' \) is not a partial truth assignment, which means that there exists some \( x \in X' \) such that \( \neg x \in X' \). This means \( X' \vdash contra^\Phi \). But since \( S \) is closed, \( contra^\Phi \subseteq S \). We show that \( unsat^\Phi \) attacks \( S \). As \( S \) was an arbitrary closed attacker of \( \{unsat^\Phi,logic^\Phi\} \), we conclude that \( \{unsat^\Phi,logic^\Phi\} \) is admissible.

We have shown that \( \Phi \) is unsatisfiable iff \( \{unsat^\Phi,logic^\Phi\} \) is an admissible extension of \( g(\Phi) \). To reduce 3-SAT-UNSAT to \( VER_{vp} \), we will construct an ABA framework that has two independent copies of \( g(\Phi) \) as defined above, in which we will be interested in verification as a complete extension of a set containing \( logic^\Phi \). For this reason, let us now prove an intermediate result that \( Def(\{logic^\Phi\}) \subseteq \{logic^\Phi\} \) iff \( \Phi \) is satisfiable.

• Suppose \( \Phi \) is satisfiable. Then, as per the argument above, there is \( X \subseteq A \) attacking \( unsat^\Phi \) such that \( \{logic^\Phi\} \) does not attack \( X \). Hence, \( unsat^\Phi \not\in Def(\{logic^\Phi\}) \). Additionally, \( \{logic^\Phi\} \) attacks \( contra^\Phi \) and does not attack itself, so \( contra^\Phi \not\in Def(\{logic^\Phi\}) \) too. Further, as every \( x \in X^\Phi \cup \neg X^\Phi \) is attacked by \( \{\neg x\} \) against which \( \{logic^\Phi\} \) does not defend \( x \), we find \((X^\Phi \cup \neg X^\Phi) \cap Def(\{logic^\Phi\}) = \emptyset \). Therefore, \( Def(\{logic^\Phi\}) \subseteq \{logic^\Phi\} \).

• Suppose now \( \Phi \) is unsatisfiable. Again, as per above, \( \{logic^\Phi\} \) defends \( unsat^\Phi \) against every closed attacker, so \( unsat^\Phi \in Def(\{logic^\Phi\}) \). Thus, \( Def(\{logic^\Phi\}) \not\subseteq \{logic^\Phi\} \).

We have just proved that \( \Phi \) is satisfiable iff \( Def(\{logic^\Phi\}) \subseteq \{logic^\Phi\} \).

Finally, given \( g(\Phi_1) \) and \( g(\Phi_2) \), define \( f(\Phi_1,\Phi_2) = (L,R,A,\neg) \) thus:

\[
\begin{align*}
L &= L_{\Phi_1} \cup L_{\Phi_2}; \\
A &= A_{\Phi_1} \cup A_{\Phi_2}; \\
R &= R_{\Phi_1} \cup R_{\Phi_2}; \\
\neg &= \text{the union of the contrary mappings of } g(\Phi_1) \text{ and } g(\Phi_2). \quad (4)
\end{align*}
\]

\( f(\Phi_1,\Phi_2) \) is constructed as the union of two independent frameworks \( g(\Phi_1) \) and \( g(\Phi_2) \), i.e. \( L_{\Phi_1} \cap L_{\Phi_2} = \emptyset \) is implicit.

Now, if \( \Phi_1 \) is satisfiable and \( \Phi_2 \) is unsatisfiable, then \( Def(\{logic^\Phi_1\}) \subseteq \{logic^\Phi_1\} \) and \( \{unsat^{\Phi_2},logic^{\Phi_2}\} \) is admissible in \( f(\Phi_1,\Phi_2) \). Thus, \( \{logic^{\Phi_1},unsat^{\Phi_2},logic^{\Phi_2}\} \) is closed and conflict-free. As \( logic^{\Phi_1} \) is also unattacked, \( \{logic^{\Phi_1},unsat^{\Phi_2},logic^{\Phi_2}\} \) is

---

4 With a slight abuse of notation we omit any sub/super-scripts on contrary mappings \( \neg \) for readability, when clear from the context how they are defined.
admissible. Further still, Def(\{unsat_Φ, logic_Φ\}) \subseteq \{unsat_Φ, logic_Φ\}, and since also Def(\{logic_Φ\}) \subseteq \{logic_Φ\}, we have that \{logic_Φ, unsat_Φ, logic_Φ\} is complete.

Conversely, if \{logic_Φ, unsat_Φ, logic_Φ\} is complete, we can likewise deduce that \{logic_Φ\} does not defend unsat_Φ, whence Def(\{logic_Φ\}) \subseteq \{logic_Φ\}, and also that \{unsat_Φ, logic_Φ\} is admissible. So Φ_1 is satisfiable and Φ_2 is unsatisfiable.

Therefore, Φ_1 is satisfiable and Φ_2 is unsatisfiable iff \{logic_Φ, unsat_Φ, logic_Φ\} is a complete extension of f(Φ_1, Φ_2).

Lastly, since f(Φ_1, Φ_2) can clearly be constructed in polynomial time, we obtain that 3-SAT-UNSAT is many-one reducible to VER_{cpl}. Consequently, VER_{cpl} is DP-hard, and so DP-c too, as required.

Veriﬁcation under admissible semantics can be dealt with similarly.

**Theorem 4.30.** For a Horn-ABA framework \(\mathcal{F}\), \(\text{VER}_{adm}\) is coNP-c.

**Proof.** First, \(\text{VER}_{adm}\) is in coNP according to Table 7 and Corollary 4.28. To prove hardness, we reduce to \(\text{VER}_{adm}\) a coNP-hard problem 3-UNSAT, which takes in input a formula Φ in 3CNF and determines if Φ is unsatisfiable. As in the proof of Theorem 4.29, consider the Horn-ABA framework \(g(\Phi)\), whereby Φ is unsatisfiable iff \{unsat_Φ, logic_Φ\} is an admissible extension of \(g(\Phi)\). This shows that 3-UNSAT is many-one reducible to \(\text{VER}_{adm}\), whence the latter is coNP-c.

For preferred semantics, we use a different reduction to prove veriﬁcation being harder, as given next.

**Theorem 4.31.** For a Horn-ABA framework \(\mathcal{F}\), \(\text{VER}_{prf}\) is \(\Pi^P_2\)-c.

**Proof.** First, \(\text{VER}_{prf}\) is in \(\Pi^P_2\) (Table 7 and Corollary 4.28). To prove hardness, we reduce 2QBF for 3CNF formulas, which is a \(\Sigma^P_2\)-c problem for determining whether a quantiﬁed Boolean formula of the form \(\exists X \forall Y \neg \Phi\) with a quantiﬁer-free 3CNF formula Φ is true (see e.g. [52]) to the complement problem of \(\text{VER}_{prf}\): for \(S \subseteq A\), determine whether \(S\) is not preferred.

For Φ = \(C_1 \land \ldots \land C_k\) with \(C_i = a \lor b \land c\) and \(X\) and \(Y\) two sets of boolean variables partitioning the variables of Φ, define \(f(\Phi, X, Y) = (L, R, A, \neg)\) thus:

- \(A = X \cup \neg X \cup Y \cup \neg Y \cup \{false, contra\}\);
- \(\forall x \in X x = \neg x, \forall x \in \neg X \neg x = x, \forall y \in Y \cup \neg Y y = y, \neg contra = false, false = true\);
- \(R = \{true \leftarrow C_1, \ldots, C_k\} \cup \{C_i \leftarrow z : z \in \{a, b, c\}, C_i = a \lor b \land c, 1 \leq i \leq k\}\)
  \(\cup\{contra \leftarrow z, \neg z : z \in X \cup Y\} \cup \{false \leftarrow x : x \in X \cup \neg X\}\).

In \(f(\Phi, X, Y)\), \(\emptyset\) is closed and hence admissible. We show that \(\emptyset\) is not a preferred extension of \((\Phi, X, Y)\) iff \(\exists X \forall Y \neg \Phi\) is true.

- If \(\emptyset\) is not preferred, then there exists an admissible extension \(S \neq \emptyset\). \(S\) is conﬂict-free, so \(S \cap (Y \cup \neg Y) = \emptyset\). \(S \neq \{contra\}\) because \(contra\) does not defend itself against \{false\}. There are now two cases.
  1. \(S = \{false\}\). In this case, \(S\) defends against any closed \(Z \subseteq A\) with \(true \in Cn(Z)\).
     Since \(S \cap (X \cup \neg X \cup Y \cup \neg Y) = \emptyset\), it must be that \(contra \in Z\). This means \(z, \neg z \in Z\)
Therefore, $\emptyset$ is not a preferred extension of $\Phi$. But this entails that there is no truth assignment in variables $X$ and $Y$ that makes $\Phi$ true (because any truth assignment has to be consistent). In other words, any truth assignment $X \cup Y$ makes $\neg \Phi$ true.

Thus, $\exists X \forall Y \neg \Phi$ holds trivially.

2. $S \neq \{false\}$. As $S$ is conflict-free, $S \neq \{false, contra\}$ either. Thus, $S \cap (X \cup \neg X) \neq \emptyset$. Then, as $S$ is closed, $false \in S$. So $S = X' \cup \{false\}$ with $X' \subseteq X$ where $X$ is some truth assignment for the variables in $X$ and $X' \neq \emptyset$. Now, for any truth assignment $Y$ for the variables in $Y$, suppose for a contradiction that $X \cup Y$ makes $\Phi$ true. Then $Z \vdash true$ for some $Z \subseteq X \cup Y$, so that $X \cup Y$ attacks $S$. As $S$ is admissible, it attacks the closure $Cn(X \cup Y) \cap A$. Note that as $X' \subseteq X$ and $X$ is a truth assignment, no element in $X$ has its contrary in $X'$. So $S$ does not attack $X$. But $S$ clearly cannot attack $Y$ either. Then it must be that $contra \in Cn(X \cup Y)$ (so that $\{false\}$ attacks $contra$), which means there is $a \in X \cup Y$ such that $\neg a \in X \cup Y$. But then $X \cup Y$ is not a truth assignment, which is a contradiction. Thus, by contradiction, $X \cup Y$ makes $\Phi$ false. Consequently, there is some truth assignment $X$ for the variables in $X$ such that for every truth assignment $Y$ for the variables in $Y$, $\neg \Phi(X, Y)$ holds.

Similarly, if there is a truth assignment $X$ such that for every assignment $Y$ for the variables in $Y$, $\neg \Phi$ is true, then $S = X \cup \{false\}$ is closed and conflict-free in $f(\Phi, X, Y)$. We show that it is also admissible, so that $\emptyset$ is not preferred. So let $A \subseteq A$ be a closed attacker of $S$. Then either $A$ attacks some element in $X$, or $A$ attacks $false$.

- If $A$ attacks some $x$ (or $\neg x$) in $X$, then $A$ contains $\neg x$ (or $x$). Then $X$ also attacks $\neg x$ (or $x$), so $S$ defends against $A$.

- Assume $A$ attacks $false$ but not $X$. Then $true \in Cn(A)$ and $A = X' \cup Y' \cup B$ where $X' \subseteq X$, $Y' \subseteq (Y \cup \neg Y)$ and $B \subseteq \{false, contra\}$. If $Y'$ were consistent, then any truth assignment extending $X \cup Y'$ would make $\Phi$ true. However, we assumed that for every assignment $Y$ for the variables $Y$, $\Phi(X, Y)$ is false. So $Y'$ is inconsistent, i.e. there is $y \in Y'$ with $\neg y \in Y'$. Thus, $contra \in Cn(A)$, and as $A$ is closed, $contra \in A$. But $contra$ is attacked by $\{false\}$, so $S = X \cup \{false\}$ defends against $A$.

Thus, $S \neq \emptyset$ defends against every attack, so is admissible, whence $\emptyset$ is not preferred.

Therefore, $\emptyset$ is not a preferred extension of $(\Phi, X, Y)$ iff $2QBF \exists X \forall Y \neg \Phi$ is true.

Since $f$ is clearly computable in polynomial time, the complement of $\text{VER}_{\text{prf}}$ is as hard as $2QBF$, i.e. $\Sigma_2^P$-c, whence $\text{VER}_{\text{prf}}$ is $\Pi_2^P$-c. \hfill \Box

Next, under stable and set-stable semantics, verification is tractable, because the general upper bounds are polynomial with a derivation oracle:

**Corollary 4.32.** For a Horn-ABA framework $\mathcal{F}$, $\text{VER}_{\text{stb}}$ and $\text{VER}_{\text{set-stb}}$ are in $P$.

**Proof.** Follows directly from Table 7 and Corollary 4.28. \hfill \Box

Lastly in terms of the verification problem, we do not yet have any hardness result under well-founded semantics in Horn-ABA and can only instantiate the general upper bound with the complexity of the derivation problem:
Corollary 4.33. For a Horn-ABA framework $\mathcal{F}$, $\text{VER}_{w-f}$ is in $\Delta^P_3$.

Proof. Follows directly from Table 7 and Corollary 4.28.

We leave it for future work to study the complexity lower bound of the verification problem under well-founded semantics. Next, we turn to credulous acceptance and existence problems and show that they are by-and-large $\Sigma^P_2$-c, with the exception of stable and set-stable semantics, where they are NP-c (we are also not yet able to establish hardness of credulous acceptance under well-founded semantics). We first tackle credulous acceptance.

Theorem 4.34. For a Horn-ABA framework $\mathcal{F}$, $\text{CA}_{adm}$, $\text{CA}_{pf}$ and $\text{CA}_{cpl}$ are $\Sigma^P_2$-c.

Proof. We know $\text{CA}_{adm}$ is in $\Sigma^P_2$. For hardness, we reduce a 2QBF problem (see proof of Theorem 4.31) to $\text{CA}_{adm}$. For $\Phi = C_1 \wedge \ldots \wedge C_k$ with $C_i = a \lor b \lor c$ and variables of $\Phi$ partitioned into $X$ and $Y$, define $f(\Phi, X, Y) = (L, R, A, \text{adm})$ thus:

- $A = X \cup \neg X \cup \neg Y \cup \{\text{false, contra}\}$;
- $\forall x \in X \ x = \neg x, \ \forall x \in \neg X \ x = x, \ \forall y \in \neg Y \ y = y, \ \overline{\text{contra}} = \text{false, } \text{false} = \text{true};$
- $R = \{\text{true} \leftarrow C_1, \ldots, C_k\} \cup \{C_i \leftarrow z : z \in \{a, b, c\}, \ C_i = a \lor b \lor c, \ 1 \leq i \leq k\}$
  $$\cup \{\text{contra} \leftarrow z, \neg z : z \in X \cup Y\} \cup \{\pi \leftarrow \text{false}\}.$$

Intuitively, a truth assignment for the variables in $X$ and $Y$ that makes $\Phi$ true will admit $\text{true}$ among its conclusions. And a truth assignment $X$ for variables in $X$ that makes $\neg \Phi$ true for any truth assignment $Y$ for variables in $Y$ will yield an admissible extension $X \cup \{\text{false}\}$ concluding $\pi$. Formally, we prove that $\exists X \ \forall Y \ \neg \Phi$ is true iff $f(\Phi, X, Y)$ has an admissible extension $S$ with $\pi \in Cn(S)$.

- If there is a truth assignment $X$ for variables in $X$ such that for every assignment $Y$ for variables in $Y$ we have $\Phi(X, Y)$ false, then $S = X \cup \{\text{false}\}$ is admissible, by a verbatim argument as in (the second bullet point of) the proof of Theorem 4.31.
- Conversely, if $S \subseteq A$ is admissible with $\pi \in Cn(S)$, then it must be that $\text{false} \in S$. Then by a verbatim argument as in (the first bullet point of) the proof of Theorem 4.31 it follows that $S = X' \cup \{\text{false}\}$ for $X' \subseteq X \cup \neg X$ such that $X'$ can be extended to a truth assignment $X$ for variables in $X$ which makes $\Phi$ false for every truth assignment $Y$ for variables in $Y$.

This ensures that $\text{CA}_{adm}$ is $\Sigma^P_2$-c just like 2QBF.

Now note that $\text{CA}_{pf}$ is equivalent to $\text{CA}_{adm}$. Indeed, if $\pi$ is credulously accepted under preferred semantics, then so is it under admissible semantics, because preferred extensions are admissible by definition. On the other hand, according to [40, Theorem 2.11(ii)], every admissible extension admits a superset preferred extension, which means that if $\pi$ is credulously accepted under admissible semantics, then so it is under preferred semantics. Thus, $\text{CA}_{pf}$ is $\Sigma^P_2$-c too. Finally, from the construction of $f(\Phi, X, Y)$, it is clear that any admissible extension $S$ as above can be extended to a complete extension (resulting into equivalence of credulous reasoning under complete and admissible semantics, like in the case of flat ABA, cf. Lemma 4.14, because non-flatness in $f(\Phi, X, Y)$ arises only when considering inconsistent assignments.) So, $\text{CA}_{cpl}$ is also $\Sigma^P_2$-c.

$\square$
The existence problem in Horn-ABA is equally as hard as the credulous acceptance problem, because the latter can be reduced to the former. In addition, the complexity lower bound for existence under well-founded semantics can also be established, due to the problem’s equivalence under well-founded and complete semantics, among others.

**Theorem 4.35.** For a Horn-ABA framework $\mathcal{F}$, $\text{EX}_{\text{adm}}$, $\text{EX}_{\text{prf}}$, $\text{EX}_{\text{cpl}}$ and $\text{EX}_{\text{w-f}}$ are $\Sigma_2^P$-c.

**Proof.** First, we know $\text{EX}_{\text{adm}}$ is in $\Sigma_2^P$. For hardness, we reduce $\text{CA}_{\text{adm}}$ to $\text{EX}_{\text{adm}}$ as follows. Define $f(\mathcal{F}, \pi) = (\mathcal{L}', \mathcal{R}', \mathcal{A}', \neg) =: \mathcal{F}'$ thus:

- $\mathcal{A}' = \mathcal{A} \cup \{b\}$ for $b \notin \mathcal{L}$;
- $\mathcal{R}' = \mathcal{R} \cup \{a \leftarrow b : a \in \mathcal{A}'\} \cup \{b \leftarrow \pi\}$.

If $\mathcal{F}$ has an admissible extension $S$ with $\pi \in Cn(S)$, then $S$ attacks $b$ in $\mathcal{F}'$, and so is an admissible extension of $\mathcal{F}'$. Conversely, if $\mathcal{F}'$ has an admissible extension $S$, then it must not contain and also must attack the self-attacking $b$, whence $S$ is admissible in $\mathcal{F}$ with $\pi \in Cn(S)$.

So $\text{CA}_{\text{adm}}$ reduces to $\text{EX}_{\text{adm}}$, whence $\text{EX}_{\text{adm}}$ is $\Sigma_2^P$-c. Then, by Lemma 4.2 and (the proofs of) Theorem 4.12 and Corollary 4.18, $\text{EX}_{\text{prf}}$, $\text{EX}_{\text{cpl}}$ and $\text{EX}_{\text{w-f}}$ are equivalent to $\text{EX}_{\text{adm}}$, and so are $\Sigma_2^P$-c too.

Both problems of existence and credulous acceptance are complete for nondeterministic polynomial time under both stable and set-stable semantics, as the following results show.

**Corollary 4.36.** For a Horn-ABA framework $\mathcal{F}$, $\text{CA}_{\text{stb}}$ is $\text{NP}$-c.

**Proof.** We know $\text{CA}_{\text{stb}}$ is in $\text{NP}$. Note that the credulous acceptance problem under stable semantics for flat LP-ABA is many-one reducible to $\text{CA}_{\text{stb}}$ (in Horn-ABA), because any flat LP-ABA framework is also a Horn-ABA framework. Since the latter problem is $\text{NP}$-c (Table 6), so is $\text{CA}_{\text{stb}}$. □

**Corollary 4.37.** For a Horn-ABA framework $\mathcal{F}$, $\text{EX}_{\text{stb}}$ is $\text{NP}$-c.

**Proof.** We know $\text{EX}_{\text{stb}}$ is in $\text{NP}$, and for hardness we can reduce $\text{CA}_{\text{stb}}$, which is $\text{NP}$-c by Corollary 4.36, to $\text{EX}_{\text{stb}}$ exactly as in the proof of Theorem 4.35. □

**Corollary 4.38.** For a Horn-ABA framework $\mathcal{F}$, $\text{EX}_{\text{set-stb}}$ and $\text{CA}_{\text{set-stb}}$ are $\text{NP}$-c.

**Proof.** First, we know both $\text{EX}_{\text{set-stb}}$ and $\text{CA}_{\text{set-stb}}$ are in $\text{NP}$ (Table 5 and Corollary 4.28). For hardness, note that both problems of existence and credulous acceptance under stable semantics in a flat LP-ABA framework reduce to $\text{EX}_{\text{set-stb}}$ and $\text{CA}_{\text{set-stb}}$, respectively, in $\mathcal{F}$, because stable and set-stable extensions coincide for flat ABA and since a flat LP-ABA framework can be seen as a Horn-ABA framework too. It then follows from Corollaries 4.37 and 4.36 that $\text{EX}_{\text{set-stb}}$ and $\text{CA}_{\text{set-stb}}$ are $\text{NP}$-c. □

Unlike with existence, but similarly to verification, we cannot yet establish the hardness of credulous acceptance under well-founded semantics, so we state the upper bound and leave further investigations for future work.

**Corollary 4.39.** For a Horn-ABA framework $\mathcal{F}$, $\text{CA}_{\text{w-f}}$ is in $\Sigma_2^P$. 28
Proof. Follows from Table 7 and Corollary 4.28. □

We finish our analysis of Horn-ABA complexity with results for sceptical acceptance. We start with the easier sceptical acceptance under stable and set-stable semantics:

Corollary 4.40. For a Horn-ABA framework \( \mathcal{F} \), \( \mathcal{S}_{\text{stb}} \) and \( \mathcal{S}_{\text{set-stb}} \) are \( \text{coNP}-\text{c} \).

Proof. We know \( \mathcal{S}_{\text{stb}} \) and \( \mathcal{S}_{\text{set-stb}} \) are \( \text{coNP} \). Note that the sceptical acceptance problem under stable semantics for flat LP-ABA is many-one reducible to \( \mathcal{S}_{\text{stb}} \) (in Horn-ABA). Since the latter problem is \( \text{coNP}-\text{c} \) (Table 6), so is \( \mathcal{S}_{\text{stb}} \). As reasoning under stable and set-stable semantics coincides in flat ABA, the same hardness result holds under set-stable semantics. □

Sceptical acceptance gets harder under admissible and complete semantics:

Theorem 4.41. For a Horn-ABA framework \( \mathcal{F} \), \( \mathcal{S}_{\text{adm}} \) and \( \mathcal{S}_{\text{cpl}} \) are \( \Pi_2^P-\text{c} \).

Proof. As a verbatim argument works for complete semantics, we give a proof for admissible semantics. First, we know \( \mathcal{S}_{\text{adm}} \) is in \( \Pi_2^P \). So for hardness, we reduce \( \mathcal{C}_{\text{adm}} \), which is \( \Sigma_2^P-\text{c} \) (Theorem 4.34), to the complement of \( \mathcal{S}_{\text{adm}} \), namely the problem of determining whether a Horn-ABA framework has an admissible extension whose conclusions do not contain a given sentence.

To this end, define \( f(\mathcal{F}) = (\mathcal{L}', \mathcal{R}', \mathcal{A}', -) =: \mathcal{F}' \) thus:

- \( \mathcal{A}' = \mathcal{A} \cup \{a, b\} \) with \( a = \pi' \) for \( a, b, \pi' \notin \mathcal{L} \);
- \( \mathcal{R}' = \mathcal{R} \cup \{a \leftarrow \, a \leftarrow b, \, B \leftarrow \pi\} \) for \( \pi' \notin \mathcal{L} \).

We next show that \( \mathcal{C}_{\text{adm}} \) can be reduced to the complement of \( \mathcal{S}_{\text{adm}} \) by showing that if one can solve the latter problem in general, and consequently for the particular pair \( (\mathcal{F}', \pi') \) of a Horn-ABA framework \( \mathcal{F}' \) and the designated sentence \( \pi' \in \mathcal{L}' \), then one can solve the former problem for the given \( \mathcal{F} \) and \( \pi \).

- On the one hand, if \( \pi' \) is not sceptically accepted in \( \mathcal{F}' \) under admissible semantics, then there is an admissible extension \( S \) of \( \mathcal{F'} \) such that \( \pi' \notin Cn'(S) \) (where \( Cn' \) is the conclusions operator for \( \mathcal{F}' \)). \( S \) must be closed, so \( a \in S \). Since \( \{b\} \) attacks \( a \), \( S \) must defend against \( \{b\} \). By construction of \( \mathcal{F}' \) it must be that \( \pi \in Cn(S) \). But note that none of the new sentences \( a, b, \pi' \) in \( \mathcal{L}' \) can be used to derive \( \pi \) in \( \mathcal{F}' \), which means that \( \pi \in Cn(S \setminus \{a\}) \). And clearly, \( S \setminus \{a\} \) is admissible in \( \mathcal{F} \). Hence, \( \pi \) is credulously accepted in \( \mathcal{F} \) under admissible semantics.

- On the other hand, suppose \( \pi \) is credulously accepted in \( \mathcal{F}' \) under admissible semantics. Then there is an admissible extension \( S \) of \( \mathcal{F} \) with \( \pi \in Cn(S) \). Clearly then, \( S \cup \{a\} \) is admissible in \( \mathcal{F}' \) and \( \pi' \notin Cn'(S \cup \{a\}) \). Thus, \( \pi' \) is not sceptically accepted in \( \mathcal{F}' \) under admissible semantics.

As \( f \) is clearly polynomial time computable, \( \mathcal{C}_{\text{adm}} \) is many-one reducible to a \( \Sigma_2^P \)-c problem, and being in \( \Pi_2^P \) is \( \Pi_2^P-\text{c} \) itself. Since a verbatim argument applies to complete semantics, the same holds for \( \mathcal{C}_{\text{cpl}} \). □

Sceptical acceptance goes a level up under preferred semantics:
Theorem 4.42. For a Horn-ABA framework \( \mathcal{F} \), \( S_{\text{ref}} \) is \( \Pi_3^P \)-c.

Proof. We know \( S_{\text{ref}} \) is in \( \Pi_3^P \). For hardness, we reduce 3QBF, which is \( \Sigma_3^P \)-hard, to the complement problem of \( S_{\text{ref}} \), namely, determining whether there exists a preferred extension \( S \) of \( \mathcal{F} \) such that \( \pi \not\in \text{Cn}(S) \). The 3QBF problem for 3CNF formulas is the decision problem taking in input a formula \( \Phi \) in CNF with 3 literals in each clause and boolean variables partitioned into sets \( X \), \( Y \) and \( Z \), and which determines if there exists one truth assignment \( \mathbf{X} \) for variables in \( X \), such that for all truth assignments \( \mathbf{Y} \) for variables in \( Y \), there is a truth assignment \( \mathbf{Z} \) for variables in \( Z \) such that \( \Phi(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) \) is true. For \( \Phi = C_1 \land \ldots \land C_k \) with \( C_i = a \lor b \lor c \) and sets \( X, Y, Z \) of boolean variables partitioning the variables in \( \Phi \), define \( f(\Phi, X, Y, Z) = (\mathcal{L}, \mathcal{R}, A, \neg) \) thus:

- \( \mathcal{A} = X \cup \neg X \cup Y \cup \neg Y \cup Z \cup \neg Z \cup \{\text{false}, \text{contra}\}; \)
- \( \forall x \in X \bar{x}, \forall \neg x \in \neg X \neg \bar{x} = x, \forall y \in Y \bar{y} = \neg y, \forall \neg y \in \neg Y \bar{y} = y, \)
- \( \forall z \in Z \cup \neg Z \bar{z} = z, \text{false} = \text{true}; \)
- \( \mathcal{R} = \{C_i \leftarrow e : e \in \{a, b, c\}, C_i = a \lor b \lor c, 1 \leq i \leq k\} \cup \{\text{true} \leftarrow C_1, \ldots, C_k\} \)
  \( \cup \{\text{contra} \leftarrow a, \neg a : a \in X \cup Y \cup Z\} \cup \{\text{false} \leftarrow y : y \in Y \cup \neg Y\} \cup \{\text{contra} \leftarrow \}. \)

Intuitively, a truth assignment for variables in \( X \) such that for all truth assignments for variables in \( Y \) there is a truth assignment for variables in \( Z \) that makes \( \Phi \) true will correspond to a preferred extension not concluding \( \text{false} \). Formally, we prove that \( \exists X \forall Y \exists Z \Phi \) is true iff \( f(\Phi, X, Y, Z) \) has a preferred extension \( S \) with \( \text{false} \not\in \text{Cn}(S) \).

- Suppose \( f(\Phi, X, Y, Z) \) has a preferred extension \( S \) with \( \text{false} \not\in \text{Cn}(S) \). Then \( S \cap (Y \cup \neg Y) = \emptyset \) and, as assumptions in \( Z \cup \neg Z \) are self-attacking, \( S \cap (Z \cup \neg Z) = \emptyset \) too. Clearly \( \text{contra} \not\in S \) either, because \( \emptyset \vdash \text{contra} \). Consequently, \( S \subseteq X \cup \neg X \). Let \( \mathbf{X} \) be a truth assignment for variables in \( X \) extending \( S \). Now, consider an arbitrary truth assignment \( \mathbf{Y} \) for variables in \( Y \). Note that as \( S \) is preferred and does not contain \( \text{false} \), \( \mathbf{X} \cup \mathbf{Y} \cup \{\text{false}\} \) is not admissible. But observe that \( \mathbf{X} \cup \mathbf{Y} \) is a truth assignment, so \( \mathbf{X} \cup \mathbf{Y} \cup \{\text{false}\} \) is conflict-free. It is also closed. So not being admissible, \( \mathbf{X} \cup \mathbf{Y} \cup \{\text{false}\} \) must be attacked by some closed \( \mathbf{A} \subseteq \mathbf{A} \) and not attack \( \mathbf{A} \) itself. Such \( \mathbf{A} \) is of the form \( X' \cup Y' \cup Z' \cup D \) with \( X' \subseteq X \cup \neg X, Y' \subseteq Y \cup \neg Y, Z' \subseteq Z \cup \neg Z \) and \( D \subseteq \{\text{false}, \text{contra}\} \). Note that if \( A \) contained \( \text{contra} \), it would be attacked by \( \emptyset \), so \( \text{contra} \not\in A \). Further, if \( A \) were attacking \( b \in \mathbf{X} \cup \mathbf{Y} \) by containing its contrary, it would be counter-attacked by \( \mathbf{X} \cup \mathbf{Y} \), so it must be that \( X' \subseteq X \) and \( Y' \subseteq Y \). Thus, \( A \) must attack \( \text{false} \), which means \( \text{true} \in \text{Cn}(A) \). Hence, \( Z' \) is a partial truth assignment for variables in \( Z \). Consequently, \( X' \cup Y' \cup Z' \) is a partial truth assignment such that \( \Phi(X', Y', Z') \) is true. So we have just shown that given a preferred extension \( S \) with \( \text{false} \not\in S \), there is a truth assignment \( \mathbf{X} \) to variables in \( X \) such that for any truth assignment \( \mathbf{Y} \) to variables in \( Y \), the fact that no proper superset of \( S \) is admissible entails that there must be always be an attacker \( A \) of \( S' = \mathbf{X} \cup \mathbf{Y} \cup \{\text{false}\} \) against which \( S' \) cannot defend, so that \( A \subseteq \mathbf{X} \cup \mathbf{Y} \cup Z \cup \neg Z \cup \{\text{false}\} \) and \( \text{true} \in \text{Cn}(A) \), meaning that there is a truth assignment \( \mathbf{Z} \) to variables in \( Z \) that makes \( \Phi \) true. That is, \( \exists X Y Z \Phi \) is true.

- Conversely, suppose there is a truth assignment \( \mathbf{X} \) for variables in \( X \) such that for all truth assignments \( \mathbf{Y} \) for variables in \( Y \) there exists a truth assignment \( \mathbf{Z} \) for variables in \( Z \) such that \( \Phi(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) \) is true. Then \( \mathbf{X} \) is admissible in \( f(\Phi, X, Y, Z) \) and note well
that false \notin X. We prove that it is actually a preferred extension. So suppose for a contradiction that it is not. Then there is an admissible \( S \subseteq A \) such that \( X \subseteq S \). It is plain that \( S \cap (Z \cup \neg Z) = \emptyset \) and contra \( \notin S \). Also, since \( X \) is a truth assignment, we find \( S \cap (X \cup \neg X) \subseteq X \). for otherwise \( S \) would attack itself. From \( X \subseteq S \) we conclude \( S \cap (X \cup \neg X) = X \). Now, suppose for a contradiction that \( Y' \subseteq S \) for some \( \emptyset \neq Y' \subseteq Y \cup \neg Y \). Then \( Y' \) must be consistent, because contra \( \notin S \). Further, being closed, \( S \) contains false. Now let \( Y \) be a truth assignment for variables in \( Y \) extending \( Y' \). Then, using our very first supposition, there is a truth assignment \( Z \) for variables in \( Z \) such that \( \Phi(X, Y, Z) \) is true. But this means \( X \cup Y \cup Z \) is closed and has true among its conclusions, so it attacks \( S \) but is not attacked by \( S \) by construction. This contradicts \( S \) being admissible, so we conclude by contradiction that \( Y' = \emptyset \), i.e. \( S \) does not contain any element from \( Y \cup \neg Y \). But then, for \( X \subseteq S \) to hold it must be that \( false \in S \) (because we already established that \( S \cap (Z \cup \neg Z) = \emptyset \) and contra \( \notin S \)). However, then using our first supposition as above, we can construct a closed set \( X \cup Y \cup Z \) of assumptions attacking, but not attacked by, \( S \). This leads to a contradiction, so \( S \) cannot be admissible. Therefore, \( X \) is preferred.

This proves that 3QBF for 3CNF formulas is many-one reducible to the complement problem of \( S_{prf} \). Thus, \( S_{prf} \) is \( \Pi^P_3 \)-c. \( \square \)

Finally, we state the instantiated upper bound regarding sceptical acceptance under well-founded semantics and leave further study on the hardness result for future work.

**Corollary 4.43.** For a Horn-ABA framework \( \mathcal{F} \), \( \mathcal{S}_{w-f} \) is in \( \Delta^P_3 \).

**Proof.** Follows from Table 7 and Corollary 4.28. \( \square \)

Table 9 (part of Table 8 in Section 1) summarises the new complexity results pertaining to Horn-ABA obtained in this section.

<table>
<thead>
<tr>
<th>sem</th>
<th>VER</th>
<th>EX</th>
<th>CA</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>adm</td>
<td>coNP-c</td>
<td>( \Sigma^P_2 )-c</td>
<td>( \Sigma^P_2 )-c</td>
<td>( \Pi^P_3 )-c</td>
</tr>
<tr>
<td>cpl</td>
<td>DP-c</td>
<td>( \Sigma^P_2 )-c</td>
<td>( \Sigma^P_2 )-c</td>
<td>( \Pi^P_3 )-c</td>
</tr>
<tr>
<td>prf</td>
<td>( \Pi^P_3 )-c</td>
<td>( \Sigma^P_2 )-c</td>
<td>( \Sigma^P_2 )-c</td>
<td>( \Pi^P_3 )-c</td>
</tr>
<tr>
<td>stb</td>
<td>P</td>
<td>NP-c</td>
<td>NP-c</td>
<td>coNP-c</td>
</tr>
<tr>
<td>set-stb</td>
<td>P</td>
<td>NP-c</td>
<td>NP-c</td>
<td>coNP-c</td>
</tr>
<tr>
<td>w-f</td>
<td>( \Delta^P_3 )</td>
<td>( \Sigma^P_2 )-c</td>
<td>( \Delta^P_3 )</td>
<td>( \Delta^P_3 )</td>
</tr>
</tbody>
</table>

Table 9: Summary of new results for classical complexity problems in Horn-ABA.

Note that Horn-ABA attains lower bounds that match the upper bounds of generic ABA under all but well-founded semantics (for which we do not have hardness results for the problems of verification, credulous and sceptical acceptance) in the sense that the complexity problems in Horn-ABA are as hard as the general upper bounds suggest, taking into account the complexity of the derivation problem. This in particular happens with the derivation problem in Horn-ABA being in P, in contrast to the derivation problem being coNP-c in Autoepistemic Logic (AEL), which also results into an
instance of ABA whose lower bounds match the general upper bounds, as discussed in the beginning of Section 4.2.2. So we have defined and studied an instance of generic ABA frameworks, namely Horn-ABA, that has a tractable derivation problem and (to the best of our efforts regarding well-founded semantics) exhibits tight lower bounds of the classical complexity problems of verification, existence, credulous and sceptical acceptance under admissible, complete, preferred, stable, set-stable and well-founded semantics.

5. PABA: Background and Problems of Interest

Having a complete picture of the complexity of ABA in general as well as its flat and potentially non-flat instances LP-ABA and Horn-ABA, we are ready to study the complexity of PABA. To this end, we first give the background on PABA in Section 5.2 (including the necessary background on abstract argumentation in Section 5.1). We then advance the function problems of interest in Section 5.3 and proceed to establish PABA complexity results in Section 6.

5.1. Abstract Argumentation (AA)

To introduce Probabilistic Assumption-Based Argumentation in Section 5.2, we need the basics of AA, which we give following [53].

An AA framework [53] is a tuple $(\text{Args}, \rightarrow)$ with

• a set \text{Args} of arguments, and
• a binary attack relation $\rightarrow$ over \text{Args}.

For $E \subseteq \text{Args}$ and $A \in \text{Args}$, $E$ defends $A$ iff for all $B \in \text{Args}$ with $B \rightarrow A$ there is $C \in E$ with $C \rightarrow B$. Then, the grounded extension of $(\text{Args}, \rightarrow)$ is $\mathcal{G} = \bigcup_{i \geq 0} G_i$, where $G_0 = \{ A \in \text{Args} \mid \exists B \in \text{Args} \text{ with } B \rightarrow A \}$ and $\forall i \geq 0$, $G_{i+1}$ is the set of arguments that $G_i$ defends. For any $(\text{Args}, \rightarrow)$, the grounded extension $\mathcal{G}$ always exists and is unique.

5.2. PABA Background

We give background on PABA following [18, 54, 30].

A PABA framework is a triple $(\mathcal{A}_P, \mathcal{R}_P, \mathcal{F})$ where:

• $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \neg)$ is an ABA framework;
• $\mathcal{A}_P \subseteq \mathcal{L}$ is a finite set of sentences, referred to as positive probabilistic assumptions, such that for each $p \in \mathcal{A}_P$ there is $\neg p \in \mathcal{L}$;\footnote{Note that $\mathcal{L}$ itself is not necessarily closed under $\neg$.} for $A \subseteq \mathcal{A}_P$, we write $\neg A = \{ \neg p : p \in A \}$ and refer to the elements of $\neg A$ as negative probabilistic assumptions;
• $\mathcal{R}_P$ is a set of probabilistic rules of the form $[p : x] \leftarrow b_1, \ldots, b_n$, where $p \in \mathcal{A}_P \cup \neg \mathcal{A}_P$, $x \in [0, 1] \cap \mathbb{Q}$ and $b_i \in \mathcal{L}$ for $i \in \{0, \ldots, n\}$;\footnote{We assume probabilities to be rational rather than real to be able to prove the complexity results for PABA. Since $\mathbb{Q}$ is dense in $\mathbb{R}$, this is not a very limiting assumption.} we adopt conventions analogous to those for rules in ABA regarding body and head of a probabilistic rules.
Put simply, a PABA framework extends a given ABA framework with probabilistic information which is expressed as statements (probabilistic assumptions) whose probabilities are derivable using the non-probabilistic information of the underlying ABA framework.

In what follows, unless stated otherwise, we assume a fixed but otherwise arbitrary PABA framework \( \mathcal{F}_P = (\mathcal{A}_P, \mathcal{R}_P, \mathcal{F}) \).

We say that \( A \subseteq \mathcal{A}_P \) is consistent iff there exists no \( p \in A \) such that \( \neg p \in A \). Then \( \mathcal{F}_P \) is well-formed iff it satisfies the following properties.

1. For a probabilistic assumption \( p \in \mathcal{A}_P \cup \neg \mathcal{A}_P \) it holds that:
   - \( p \notin A \);
   - \( p \neq \text{head}(r) \) for any \( r \in \mathcal{R} \);
   - \( [p : x] \notin \text{body}(r) \) for any \( r \in \mathcal{R} \cup \mathcal{R}_P \).

2. For each probabilistic rule \( [p : x] \leftarrow b_1, \ldots, b_n \) in \( \mathcal{R}_P \), it holds that \( [\neg p : 1 - x] \leftarrow b_1, \ldots, b_n \) is also a probabilistic rule in \( \mathcal{R}_P \).

3. For each \( p \in \mathcal{A}_P \), there exists \( B_p \subseteq \mathcal{A}_P \) such that for each \( \subseteq \)-maximally consistent subset \( \{b_1, \ldots, b_n\} \) of \( B_p \cup \neg B_p \), there exists a probabilistic rule \( [p : x] \leftarrow b_1, \ldots, b_n \) in \( \mathcal{R}_P \).

4. For \( r_1, r_2 \in \mathcal{R}_P \), with \( \text{head}(r_1) = [p : x] \) and \( \text{head}(r_2) = [p : y] \) such that \( x \neq y \), one of the following holds:
   - \( \text{body}(r_1) \subseteq \text{body}(r_2) \) or \( \text{body}(r_2) \subseteq \text{body}(r_1) \);
   - there exists \( q \in \mathcal{A}_P \) such that \( q \in \text{body}(r_1) \) and \( \neg q \in \text{body}(r_2) \).

Henceforth, \( \mathcal{F}_P \) is assumed to be well-formed.

Intuitively, we are interested in PABA frameworks where 1. probabilistic information can be inferred only using probabilistic rules; 2. complementary probabilities of probabilistic assumptions are derivable using probabilistic rules; 3-4. probabilistic statements can always be consistently and coherently inferred. Such PABA frameworks encode Bayesian Networks (BN) in the sense that each pair \( p, \neg p \) of probabilistic assumptions corresponds to truth assignments of variable \( p \) in a BN, and probabilistic rules correspond to conditional probability tables [54].

**Example 5.1.** Recall Example 3.1 and the answers to credulous and sceptical acceptance problems given in Example 3.2. We believe that conclusions in Example 3.2 are unsatisfactory due to the lack of probabilistic modelling of the information, e.g. how likely it is that a given patient has complications, or the likelihood of the effectiveness of an action. Therefore, revisiting Example 3.1, we make the rule \( E \leftarrow A \) in ABA conditional on the probabilistic effectiveness assumption \( eff \) pertaining to action \( A \), as in Example 1.1, and we also incorporate probabilistic information concerning the complications rate due to exacerbations \( exacrb \). Thus, we formalise the situation in \( \mathcal{F}_P = (\mathcal{A}_P, \mathcal{R}_P, \mathcal{F}) \) with

- \( \mathcal{F} \) given by

---

7 We assume that \( \neg \neg p = p \) for all \( \neg p \in \neg \mathcal{A}_P \)
- $A = \{A, A', \text{no compl}\}$,
- $R = \{E \leftarrow A, \text{eff}, E' \leftarrow A', \text{no compl}, \text{no compl} \leftarrow \text{exacr}\}$,
- $\mathcal{A} = A'$, $\overline{A} = A$.

- $\mathcal{A}_p = \{\text{eff, exacr}\}$,
- $\mathcal{R}_p = \{[\text{eff} : 0.7] \leftarrow, [\neg \text{eff} : 0.3] \leftarrow, [\text{eff} : 0.2] \leftarrow \text{exacr}, [\neg \text{eff} : 0.8] \leftarrow \text{exacr}, [\text{exacr} : 0.1] \leftarrow, [\neg \text{exacr} : 0.9] \leftarrow \}.

$\mathcal{F}_p$ is well-formed. In $\mathcal{F}_p$ the acceptance of e.g. effect $E$ comes with a probability, which is arguably more intuitive than its crisp (non-)acceptance as in Example 3.2.

The probability space in PABA is defined via a probability distribution over possible worlds. A possible world is a $\subseteq$-maximally consistent subset $w \subseteq A_P \cup \neg A_P$. Intuitively, a possible world determines an uncertain situation at hand in which we use argumentation to reason about the situation. In what follows, $W$ denotes the set of all possible worlds. To define the probability $P(w)$ of $w \in W$, we will use an AA framework and its grounded extension to determine which probabilistic arguments constructed using probabilistic assumptions in $w$ are acceptable in the AA framework and should thus contribute to $P(w)$. Given $w \in W$, we define an AA framework (see Section 5.1) $P_w = (\text{Args}, \rightarrow)$ thus:

- Args is the set of arguments of the form $A \vdash \phi$ for $A \subseteq A$ in the ABA framework $(\mathcal{L}, \mathcal{R}_p \cup \mathcal{R}_w, \mathcal{A}, \neg)$, where
  - $\mathcal{R}_w = \mathcal{R} \cup \{p \leftarrow : p \in w\}$, and
  - an argument $A = A \vdash \phi$ in Args is called:
    - * probabilistic if $\phi = [p : x]$ for some $p \in A_P \cup \neg A_P$;
    - * non-probabilistic otherwise.
- For $A = A \vdash \phi$ and $B = B \vdash \psi$ in Args, we have $A \rightarrow B$ iff one of the following holds:
  a) $A$ is a non-probabilistic argument and $\phi = \overline{b}$ for some $b \in \text{Cl}(B)$;
  b) $A = \emptyset$, $\phi = p$ for some $p \in A_P \cup \neg A_P$ and $B$ is a probabilistic argument with $\psi = [-p : x]$ for some $x \in [0, 1]$;
  c) $A$ and $B$ are probabilistic arguments with $\phi = [p : x]$ and $\psi = [p : y]$ for some $p \in A_P \cup \neg A_P$, such that $\text{body}(r_B) \subseteq \text{body}(r_A)$, where $A$ (as a tree) has root $\phi$ with children labelled by elements of $\text{body}(r_A)$ and $B$ (as a tree) has root $\psi$ with children labelled by elements of $\text{body}(r_B)$.

(In this case, we say that $A$ attacks $B$ by specificity.)

Remark 1. Note well that in all of the above we allow the underlying ABA framework $\mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \neg)$ of the given PABA framework $\mathcal{F}_p = (\mathcal{A}_p, \mathcal{R}_p, \mathcal{F})$ to be non-flat. While Dung and Thang defined PABA only with respect to flat ABA [18], and the same setting was kept by Hung in [30], we do not restrict ourselves to flat ABA frameworks and hereby generalise the definition of PABA.

Intuitively, arguments in $P_w$ are as usual in ABA, but possibly with probabilistic statements as conclusions. Accordingly, attacks are extended with respect to the probabilistic information so that b) certain information trumps over complementary uncertain information, and c) more specific arguments trump over less specific ones. The
AA framework associated with the given possible world thus allows to reason about the acceptable probabilistic as well as non-probabilistic information in that world, and thence to determine the probability of the world. Acceptability will be interpreted with respect to grounded semantics – intuitively, semantics that leaves no room for doubt.

**Example 5.2.** [Example 5.1 continued] There are four possible worlds in $F_p$ from Example 5.1: 1. $w_1 = \{\text{eff}, \neg \text{exacrb}\}$, 2. $w_2 = \{\neg \text{eff}, \text{exacrb}\}$, 3. $w_3 = \{\text{eff}, \text{exacrb}\}$, 4. $w_4 = \{\neg \text{eff}, \neg \text{exacrb}\}$. These yield four AA frameworks $P_{w_i}$ for $i \in \{1, \ldots, 4\}$, with various arguments and attacks. For instance, in $P_{w_1} = (\text{Args}_{w_1}, \succeq)$ we have

- $\text{Args}_{w_1} = \{\{A\} \vdash A, \{A'\} \vdash A', \{\text{no compl}\} \vdash \text{no compl}, \{A', \text{no compl}\} \vdash E',
\{A\} \vdash E, \{A\} \vdash \overline{A}, \{A'\} \vdash \overline{A}, \emptyset \vdash \text{eff}, \emptyset \vdash \neg \text{exacrb},
\emptyset \vdash [\text{exacrb} : 0.1], \emptyset \vdash [\neg \text{exacrb} : 0.9], \emptyset \vdash [\text{eff} : 0.7], \emptyset \vdash [\neg \text{eff} : 0.3]\},$
- $\{A\} \vdash \overline{A} \succeq \{A'\} \vdash A; \{A\} \vdash \overline{A} \succeq \{A', \text{no compl}\} \vdash E';
\{A'\} \vdash E; \emptyset \vdash \neg \text{exacrb} \succeq \emptyset \vdash [\text{exacrb} : 0.1]; \emptyset \vdash \text{eff} \succeq \emptyset \vdash [\neg \text{eff} : 0.3].$

In $P_{w_2} = (\text{Args}_{w_2}, \succeq)$ we have two more probabilistic arguments in contrast to those in $\text{Args}_{w_1}$, namely $\emptyset \vdash [\text{eff} : 0.2]$ and $\emptyset \vdash [\neg \text{eff} : 0.8]$, due to the presence of the rule $\text{exacrb} \leftarrow \text{eff}$. Importantly, then, we find the attack $\emptyset \vdash [\neg \text{eff} : 0.8] \succeq \emptyset \vdash [\neg \text{eff} : 0.3]$ by specificity, among others.

The other two AA frameworks can be similarly constructed.

**Remark 2.** Note that one in principle could reason directly in the ABA framework $(\mathcal{L}, \mathcal{R}_p \cup \mathcal{R}_w, \mathcal{A}, \succeq)$ itself rather than in the resulting AA framework $P_w = (\text{Args}, \succeq)$. However, due to existence of probabilistic rules, that would require redefining ABA attacks and semantics accordingly. Instead, the attacks are more readily defined at the abstract level among arguments in $P_w$, and the semantics need not be modified at all.

Given the AA framework $P_w$ built from the arguments of the ABA framework $(\mathcal{L}, \mathcal{R}_p \cup \mathcal{R}_w, \mathcal{A}, \succeq)$ using probabilistic rules as well as probabilistic assumptions from the possible world $w$ as facts, the grounded extension of $P_w$ consisting of the most well argumentatively defensible yet uncertain information allows to determine the probability of $w$. Formally, the probability $P(w)$ of a possible world $w \in \mathcal{W}$ is defined as the product of probabilities of probabilistic assumptions in $w$ for which probabilistic arguments are in the grounded extension of $P_w$:

$$P(w) = \prod_{\mathcal{G} \text{ the grounded extension of } P_w; A' \vdash [p,x] \in \mathcal{G}; \ p \in w} x.$$  

(By convention, if $\mathcal{G}$ is empty, then the empty product is equal to 1.)

**Example 5.3.** [Example 5.2 continued] We can determine that the grounded extension $\mathcal{G}_1$ of $P_{w_1}$ from Example 5.2 contains two probabilistic arguments $\emptyset \vdash [\neg \text{exacrb} : 0.9]$ and $\emptyset \vdash [\text{eff} : 0.7]$, whence $P(w_1) = 0.9 \cdot 0.7 = 0.63$. Similarly, the only probabilistic arguments contained in the grounded extension $\mathcal{G}_2$ of $P_{w_2}$ are $\emptyset \vdash [\neg \text{eff} : 0.8]$ and $\emptyset \vdash [\text{exacrb} : 0.1]$, so that $P(w_2) = 0.08$. In a similar fashion, we can determine $P(w_3) = 0.02$ and $P(w_4) = 0.27$. Note that $\sum_{i=1}^{4} P(w_i) = 1$.  

35
Having the probabilities of possible worlds, we use ABA to reason about the consequences of the non-probabilistic information under various acceptability semantics. Formally, given \( w \in \mathcal{W} \), standard ABA semantics (see Section 3.1) determines the extensions of an ABA framework associated to \( w \) as follows:

- let \( \mathcal{F}_w = (\mathcal{L}, \mathcal{R}_w, A, \neg) \) be an ABA framework with
  \[ \mathcal{R}_w = \mathcal{R} \cup \{ p \leftarrow : p \in w \} \];
- for \( \text{sem} \in \{ \text{adm}, \text{prf}, \text{stb}, \text{set-stb}, \text{cpl}, \text{w-f} \} \), we write \( \mathcal{F}_w \models \text{sem} S \) to denote that \( S \subseteq A \) is a \text{sem} extension of \( \mathcal{F}_w \).

Example 5.4. [Example 5.2 continued] For \( w_1 \in \mathcal{W} \) from Example 5.2 we have the associated ABA framework \( \mathcal{F}_{w_1} = (\mathcal{L}_1, \mathcal{R}_1, A, \neg) \) with

- \( A = \{ A, A', \text{no\_compl} \} \),
- \( \mathcal{R}_1 = \{ E \leftarrow A, \text{eff}, E' \leftarrow A', \text{no\_compl}, \text{no\_compl} \leftarrow \text{exacrb}, \text{eff} \leftarrow , \neg\text{exacrb} \leftarrow \} \),
- \( \overline{A} = A', \overline{A} = A \).

We find e.g. \( \mathcal{F}_{w_1} \models \text{stb} \{ A, \text{no\_compl} \} \) and \( \mathcal{F}_{w_1} \models \text{stb} \{ A', \text{no\_compl} \} \).

Henceforth, \( \mathcal{F}_w \) is the ABA framework associated to \( w \in \mathcal{W} \).

The semantics of PABA is defined thus: the probability \( P_{\text{sem}}(\pi) \) that \( \pi \in \mathcal{L} \) is accepted under \text{sem} semantics is

\[
P_{\text{sem}}(\pi) = \sum_{w \in \mathcal{W}; \; \mathcal{F}_w \models \text{sem} S \; \pi \in \text{Cn}(S)} P(w).
\]

In other words, to find the probability of a sentence \( \pi \) to be accepted under given semantics, we sum the probabilities of the possible worlds \( w \) for which \( \pi \) is credulously accepted (i.e. conclusion of at least one extension) under the given semantics in the associated non-probabilistic ABA framework \( \mathcal{F}_w \).

Example 5.5. Combining and extending the illustrated results from Examples 5.2, 5.3 and 5.4, we could find e.g. \( P_{\text{stb}}(E) = P(w_1) + P(w_3) = 0.65 \).

Overall, reasoning in PABA goes as follows. Given a possible world \( w \), i.e. a \( \subseteq \)-maximally consistent set of probabilistic assumptions, consider AA framework \( P_w \) with arguments built from the elements of \( w \) as facts and probabilistic rules, as well as the assumptions and rules of the ABA framework \( \mathcal{F} \) underlying \( \mathcal{F}_P = (A_P, \mathcal{R}_P, \mathcal{F}) \). Determine the grounded extension of \( P_w \) in order to establish the acceptable probabilistic assumptions among those in \( w \), whose probabilities multiplied then yield the probability \( P(w) \) of \( w \). Knowing the probabilities over possible worlds \( \mathcal{W} \), for any such \( w \in \mathcal{W} \) consider the associated (non-probabilistic) ABA framework \( \mathcal{F}_w \) with elements of \( w \) as facts and determine extensions under any given semantics. To obtain the probability of any one sentence \( \pi \) from \( \mathcal{F} \) to be accepted in \( \mathcal{F}_P \), sum the probabilities of the possible worlds in which \( \pi \) is credulously concluded under given semantics.

Finally, as in [30], we only consider PABA frameworks that make sense probabilistically, i.e., satisfy the coherence property [18] that \( \sum_{w \in \mathcal{W}} P(w) = 1 \). A class of such frameworks is defined as follows.
• The dependency graph of \((A_P, R_P, F)\) is a directed graph with
  - nodes from \(L\) with \(p\) and \(\neg p\) merged for each \(p \in A_P\), and
  - edges of the form \((a, b)\) iff one of the following holds:
    * \(a \in A\) and \(b = a\);
    * there is \(r \in R\) with \(\text{head}(r) = a\) and \(b \in \text{body}(r)\);
    * there is \(r \in R_P\) with \(\text{head}(r) = [a : x]\) and \(b \in \text{body}(r)\).

• \(F_P\) is probabilistically acyclic iff there is no infinite path starting from a probabilistic assumption in the dependency graph of \(F_P\).

Example 5.6. \(F_P\) from Example 5.1 is probabilistically acyclic with \((\text{eff}, \text{exacrb})\) being the only path that starts from a probabilistic assumption in the dependency graph of \(F_P\). Consequently, and as noted in Example 5.3, \(F_P\) satisfies the coherence property.

In the rest, \(F_P\) is (well-formed and) probabilistically acyclic.

5.3. Probabilistic Complexity Problems in PABA

We introduce the probabilistic counterparts of the problems given in Section 3.4. These will be the complexity problems for PABA of interest in this paper.

Definition 5.1. Let \(F = (A_P, R_P, F)\) be a PABA framework with the underlying ABA framework \(F = (L, R, A, \ldots)\) and \(\text{sem} \in \{\text{adm}, \text{prf}, \text{set-stb}, \text{cpl}, \ldots\}\).

• The probabilistic set verification problem for a given \(S \subseteq A\), \(P_{\text{VER}}\text{sem}(S)\), is the problem of computing the probability over possible worlds \(w \in W\), that \(S\) is a \(\text{sem}\) extension of ABA framework \(F_w\).
  I.e. \(P_{\text{VER}}\text{sem}(S)\) amounts to determining
  \[
  \sum_{w \in W} P(w). \text{ w \in W} \text{ where } F_w \vdash \text{sem} S.
  \]

• The probabilistic credulous acceptance problem for a given \(\pi \in L\), \(P_{\text{CA}\text{sem}}(\pi)\), is the problem of computing the probability over possible worlds \(w \in W\), that \(\pi\) is credulously accepted under \(\text{sem}\) semantics, in other words that there exists a \(\text{sem}\) extension \(S \subseteq A\) of ABA framework \(F_w\) such that \(\pi \in Cn(S)\).
  I.e. \(P_{\text{CA}\text{sem}}(\pi)\) amounts to determining
  \[
  \sum_{w \in W} P(w). \exists S \subseteq A : F_w \vdash \text{sem} S, \pi \in Cn(S).
  \]

• The probabilistic sceptical acceptance problem for a given \(\pi \in L\), \(P_{\text{SA}\text{sem}}(\pi)\), is the problem of computing the probability over possible worlds \(w \in W\), that \(\pi\) is sceptically accepted under \(\text{sem}\) semantics, in other words that for each \(\text{sem}\) extension \(S \subseteq A\) of ABA framework \(F_w\) it holds that \(\pi \in Cn(S)\).
  I.e. \(P_{\text{SA}\text{sem}}(\pi)\) amounts to determining
  \[
  \sum_{w \in W} P(w). \forall S \subseteq A (F_w \vdash \text{sem} S \rightarrow \pi \in Cn(S)).
  \]

We next study the complexity of these problems in PABA.
6. Complexity Results for PABA

We here give complexity results for PABA. We establish that all three probabilistic problems of set verification (P-VER), credulous (P-CA) and sceptical (P-SA) acceptance have upper bounds in \( \text{FP}^\text{WP} \) for PABA in general, and are \( \text{FP}^\text{WP} \)-c for PABA instantiated with Horn-ABA, under every semantics considered in this paper, in either flat or possibly non-flat cases. We first consider the probabilistic set verification problem in Section 6.1 and then turn to probabilistic credulous and sceptical acceptance problems in 6.2. After establishing results for the probabilistic problems given in Section 5.3, we propose and study new decision problems for PABA in Section 6.2.1.

We recall that \( \mathcal{F}_p = (\mathcal{A}_p, \mathcal{R}_p, \mathcal{F}) \) is a well-formed, probabilistically acyclic PABA framework with the underlying potentially non-flat ABA framework \( \mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \neg) \). For a possible world \( w \in \mathcal{W} = \{ w \subseteq \mathcal{A}_p \cup \neg \mathcal{A}_p : w \text{ is } \subseteq \text{-maximally consistent} \} \), the probability of \( w \) is given by \( P(w) = \prod_{\mathcal{G} \text{ grounded in } P_w : \mathcal{A} \vdash [p,x] \in \mathcal{G} ; p \in w} x \), where \( P_w \) is the AA framework constructed using arguments of the ABA framework \( (\mathcal{L}, \mathcal{R}_p \cup \mathcal{R}_w, \mathcal{A}, \neg) \) with \( \mathcal{R}_w = \mathcal{R} \cup \{ p \leftarrow : p \in w \} \). Furthermore, \( \mathcal{F}_w \) is the associated ABA framework \( (\mathcal{L}, \mathcal{R}_w, \mathcal{A}, \neg) \) and \( \mathcal{F}_w \vdash_{\text{sem}} S \) denotes that \( S \subseteq \mathcal{A} \) is a sem extension of \( \mathcal{F}_w \) for \( \text{sem} \in \{ \text{adm, prf, stb, set-stb, cpl, w-f} \} \). Throughout, we assume that the ABA derivation problem belongs to \( \text{PH} \), the Polynomial Hierarchy. This is a reasonable assumption since, for instance, \( \text{PH} \) is equal to the set of boolean queries expressible in second-order logic.\(^8\)

6.1. Probabilistic Set Verification in PABA

To find the complexity of the problems we are interested in for PABA, we first prove that computing the probability of a possible world is in the polynomial hierarchy.

**Lemma 6.1.** Computing the probability \( P(w) \) of a possible world \( w \in \mathcal{W} \) is in \( \text{PH} \).

**Proof.** Note that in general, constructing \( P_w \) can be exponential in time and space. We thus describe a procedure to compute \( P(w) \) in polynomial time with the use of PH-oracles without constructing the full \( P_w \) and determining its grounded extension \( \mathcal{G} \). Specifically, we exploit the well-formedness properties of \( \mathcal{F}_p \) and the structure of \( P_w \) to analytically show that the probabilistic arguments \( A \vdash [b : x] \) in \( P_w \) that would appear in \( \mathcal{G} \) are of a certain form and that their supporting sets of assumptions are contained in the well-founded extension of \( \mathcal{F}_w \). Then, for the probabilistic arguments relevant for computing \( P(w) \), we can determine whether they would be in \( \mathcal{G} \) by only using PH-oracles to query about derivations in \( \mathcal{F}_w \). We proceed to prove this by first showing how PH-oracles suffice to determine the (probabilistic rules used in the) probabilistic arguments in question, and the by showing how to find the ones needed to establish \( P(w) \).

Fix a probabilistic assumption \( b \in w \). We first show that for every probabilistic rule \( [b : x] \leftarrow a_1, \ldots, a_n \) in \( \mathcal{R}_p \) we can use a PH-oracle to determine whether there exists an argument \( B \vdash [b : x] \) in \( P_w \) such that the set \( B \subseteq \mathcal{A} \) of assumptions deriving \( [b : x] \) in \( \mathcal{F}_w \).

---

\(^8\)https://people.cs.umass.edu/~immerman/descriptive_complexity.html
is contained in the well-founded extension $G_{F_w}$ of $F_w$. (We will show later in the proof what happens if $G_{F_w}$ does not exist.) Consider an oracle implementing the following algorithm:

a) Guess $B \subseteq A$.

b) Check if $B \vdash a_i$ in $F_w$, for every $a_i \in \{a_1, \ldots, a_n\}$.

c) Check if $B \subseteq G_{F_w}$, where $G_{F_w}$ is the well-founded extension of $F_w$.

Since we assumed that the derivation problem in ABA belongs to $PH$, step b) above can be done with $n$ calls to a $PH$-oracle. Moreover, by Corollary 4.20, constructing $G_{F_w}$ takes $|A|$ calls to a $PH$-oracle, so step c) can be done with a $PH$-oracle too. The oracle executing the above algorithm therefore operates in $NP^{PH} = PH$.

Now, if checks b) and c) are positive, then there exists an argument $B \vdash [b : x]$ in $P_w$ such that $B \subseteq G_{F_w}$. So, for a given $b \in w$, we can go through the rules in $R_P$ and determine the set

$$R_b = \{[b : x] \leftarrow a_1, \ldots, a_n \in R_P : \exists \text{ argument } B \vdash [b : x] \text{ in } P_w \text{ with } B \subseteq G_{F_w}\}$$

in polynomial time using calls to oracles in $PH$.

(To facilitate understanding, it may be instructive to exemplify some objects with the help of Example 5.2. There, consider $w_2 = \{\neg eff, exacrb\}$ and set $b = \neg eff$. From $R_P$ given in Example 5.1, we will find $R_{\neg eff} = \{\neg eff : 0.3\} \leftarrow \{\neg eff : 0.8\} \leftarrow exacrb$ with arguments $A = \emptyset \vdash \{\neg eff : 0.3\}$ and $B = \emptyset \vdash \{\neg eff : 0.8\} \leftarrow exacrb \leftarrow \mathbf{exacrb}$.

-Trivially, $\emptyset \subseteq G_{F_{w_2}}$.)

We next establish that we can deterministically find a rule in $R_b$ and a corresponding argument $B = B \vdash [b : x]$ using that rule such that $B$ belongs to the grounded extension $G$ of $P_w$. To this end, we first show that $R_b$ is not empty and then how to pick a rule in $R_b$ that yields such $B$.

Since $F_P$ is well-formed (see Section 5.2), property 3 guarantees that there is $B_b \subseteq A_P$ such that for each $\subseteq$-maximally consistent subset $\{b_1, \ldots, b_m\}$ of $B_b \cup \neg B_b$, there is a probabilistic rule $[b : x] \leftarrow b_1, \ldots, b_m$ in $R_P$. Let $\{p_1, \ldots, p_m\} = w \cap (B_b \cup \neg B_b)$. Then $\{p_1, \ldots, p_m\}$ is $\subseteq$-maximally consistent, because $w$ is. So $[b : x] \leftarrow p_1, \ldots, p_m \in R_P$. But note that for $1 \leq i \leq m$ we have $p_i \leftarrow \in R_w$ by definition of $R_w$. Consequently, $\emptyset \vdash [b : x]$ in $F_w$, and so $\emptyset \vdash [b : x]$ is an argument in $P_w$. Further, $\emptyset \subseteq G_{F_w}$, and therefore $R_b$ is not empty.

Now, find $r_b = [b : x] \leftarrow b_1, \ldots, b_n \in R_b$ such that $\text{body}(r_b)$ is $\subseteq$-maximal among rules in $R_b$. Let $B = B \vdash [b : x]$ be an argument in $P_w$ with the “last rule” (whose body consists of elements labelling the children of $[b : x]$ in $B$ as a tree) being $r_b$. Then $B \subseteq G_{F_w}$.

Consider now the three types of attacks a), b) and c) that are possible in $P_w$ (see Section 5.2). Let $A = A \vdash \varphi$ be a potential attacker of $B$ in $P_w$.

a) $A$ is a non-probabilistic argument and $\varphi = \overline{a}$ for some $a \in B$. Since $B \subseteq G_{F_w}$, it is defended in $F_w$ by $G_{F_w}$. This means that every attack of type a) against $B$ is defended by arguments in $P_w$ that are supported by subsets of $G_{F_w}$ and hence belong to $G$.

39
b) A is a non-probabilistic argument with \( \varphi = \neg b \). By property 1 of well-formedness, there is no rule in \( R \) with \( \neg b \) as head. So it must be that \( \neg b \not\in R_w \), meaning \( \neg b \in w \), which makes \( w \) inconsistent (recall that \( b \in w \) to begin with). So \( B \) cannot be attacked in \( P_w \) by attack of type b).

c) A is a probabilistic argument with \( \varphi = [b : y] \) with its “last rule” \( r_A \) such that \( \text{body}(r_b) \subseteq \text{body}(r_A) \) so that A attacks B by specificity, i.e. as attack of type c). Since \( \text{body}(r_b) \) is \( \subseteq \)-maximal among rules in \( R_b \), it must be that \( r_A \not\in R_b \). In particular, \( A \not\subseteq G_{F_w} \). Note that by definition of \( P_w \), to be able to construct \( A, r_A \) must satisfy \( \text{body}(r_A) \cap (A_P \cup \neg A_P) \subseteq w \), due to property 1 of well-formedness (i.e. because probabilistic assumptions can only be derived in \( (L, R_P \cup R_w, A, \neg) \) as facts using \( R_w \)). Thus, \( \text{body}(r_A) \cap (A_P \cup \neg A_P) \subseteq w \), and hence \( \text{body}(r_A) \cap (A_P \cup \neg A_P) \subseteq G_{F_w} \). This means \( \emptyset \not\in A \cap A \not\subseteq G_{F_w} \), i.e. that A is supported by some (standard) assumptions \( A' := A \setminus G_{F_w} \subseteq A \) that are not in the well-founded extension of \( F_w \).

We now show that B must therefore be defended from A by arguments in \( P_w \) that are supported by subsets of \( G_{F_w} \) and hence belong to \( G \). Indeed, since \( A' \not\subseteq G_{F_w} \), in \( F_w \) there exists a closed attacker \( A_1 \subseteq A \) of \( A' \) such that \( G_{F_w} \) does not attack \( A_1 \); in detail, there exists a complete extension \( C \) such that \( A' \not\subseteq C \), meaning C does not defend \( A' \), meaning there is a closed \( A_1 \subseteq A \) such that \( A_1 \) attacks \( A' \) but C does not attack \( A_1 \).

- i. If \( A_1 \) is the closed attacker of \( G_{F_w} \) and hence \( \emptyset \) \( \subset \)\( A \) contains \( A_1 \), hence \( G_{F_w} \) contains \( A_1 \) and thus attacks \( A' \), so that \( G_{F_w} \) defends B from A in \( F_w \) and hence \( G \) defends B from A in \( P_w \).
  - ii. Else, \( A_1 \) is not the closed attacker of \( G_{F_w} \). Then there is a closed attacker \( A_2 \) of \( A_1 \) such that \( G_{F_w} \) does not attack \( A_2 \). If \( G_{F_w} \) defends (and thus contains) \( A_2 \), then \( G_{F_w} \) attacks \( A_1 \), which contradicts the complete extension C not attacking \( A_1 \). Hence, it must be that \( G_{F_w} \) does not defend \( A_2 \). So \( A_2 \) is like \( A_1 \): neither attacked nor defended by \( G_{F_w} \). Since \( F \) and all of its components are assumed to be finite (see Section 3.1), continuing in this manner we will find a cycle of attacks in \( F_w \), including \( A' \). But this contradicts \( F \) being probabilistically cyclic (see just before Example 5.6): in the dependency graph of \( F \), there is an infinite path starting from the probabilistic assumption \( b \) (appearing in \( \text{head}(r_A) = [b : y] \)) going through the proof tree of A, reaching some \( a \in A' \subseteq A \) and eventually looping through the identified cycle of attacks. Therefore, by contradiction, we must be in case i. above.

We have thus showed that \( B \) is defended in \( P_w \) from A by \( G \).

Consequently, B belongs to the grounded extension \( G \) of \( P_w \).

(Continuing with the illustration on Example 5.2, where \( R_{\text{eff}} = \{ [-\text{eff} : 0.3] \leftarrow , [-\text{eff} : 0.8] \leftarrow \text{exacrb} \} \), we have \( r_{\text{eff}} = [-\text{eff} : 0.8] \leftarrow \text{exacrb} \). There, argument \( B = \emptyset \vdash [-\text{eff}:0.8]\rightarrow \text{exacrb} \). Thus \( [-\text{eff} : 0.8] \) is not attacked and in the grounded extension \( G_2 \) of \( P_w \), as stated in Example 5.3.)

This shows that given \( b \in w \) we can find \( x \) such that there exists \( B \vdash [b : x] \in G \) in polynomial time with calls to PH-oracles, assuming existence of the well-founded extension \( G_{F_w} \) of \( F_w \). By doing this for every \( b \in w \) we compute \( P(w) \) in polynomial time with calls to PH-oracles, under the same assumption. So we finally show that we
can dispense with the assumption of existence of $G_{\mathcal{F}}$, by showing what has to happen for $G_{\mathcal{F}}$, not to exist and that in that case computing $P(w)$ is trivial.

So suppose $\mathcal{F}$ does not have a well-founded extension. Then, by definition, $\mathcal{F}$ has no complete extension, and hence no preferred extension either (since preferred extensions are complete). Then, by [40, Theorem 2.12(ii)], $\emptyset$ is not closed in $\mathcal{F}$. Hence, by [40, Theorem 2.12(i)], $\mathcal{F}$ has no admissible extension either. In particular, the closure $Cl(\emptyset)$ of the empty set is not admissible in $\mathcal{F}$. Since $Cl(\emptyset)$ is closed, this entails that it either is not conflict-free or does not defend itself. On the one hand, if $Cl(\emptyset)$ does not defend itself, then there is a closed set $A \subseteq \mathcal{A}$ that attacks $Cl(\emptyset)$ but $Cl(\emptyset)$ does not attack $A$. So there must be $a \in A \cap Cl(\emptyset)$ such that $A \vdash \pi$. Note that $Cl(\emptyset) \subseteq A$, since $A$ is closed. So $A$ attacks itself too. On the other hand, if $Cl(\emptyset)$ is not conflict-free, then likewise there must be $a \in A \cap Cl(\emptyset)$ such that $Cl(\emptyset) \vdash \pi$.

Now, consider $P_w$. Then, as above, $Cl(\emptyset)$ does not defend itself in $\mathcal{F}$ against some self-attacking $A \vdash \pi$ (with $A$ closed, possibly $A = Cl(\emptyset)$). Thus, for any $w \in \mathcal{W}$, all arguments of $P_w$ are attacked by $A = A \vdash \pi$. This entails in particular that the grounded extension $\mathcal{G}$ of $P_w$ is empty. This means $P(w) = 1$, whence $\sum_{w \in \mathcal{W}} P(w) > 1$ whenever $|\mathcal{W}| > 1$. By definition of $\mathcal{W}$, it holds that $|\mathcal{W}| \geq 1$. Therefore, for $\mathcal{F}_P$ to satisfy the coherence property that $\sum_{w \in \mathcal{W}} P(w) = 1$ (see Section 5.2), it must be that $\mathcal{W} = \{\emptyset\}$. But then trivially $P(\emptyset) = 1$, and this can be computed trivially in polynomial time by checking if $\mathcal{A}_P = \emptyset$ to begin with.

It thus follows from Lemma 6.1 that in PABA we do not need to construct $P_w$ to determine the probability of a possible world and it suffices to have a PH-oracle instead. This fact will be used in the proofs to follow.

We next find a general upper bound for probabilistic set verification in PABA, and then show the bound to be strict (i.e. a lower bound too) for PABA instantiated with Horn ABA.

**Theorem 6.2 (P-VER Upper Bounds).** For a generic PABA framework $\mathcal{F}_P$ and $sem \in \{adm, prf, stb, set-stb, cpl, w-f\}$, P-VER$_{sem}$ is in FP$^{#PH}$.

**Proof.** First, an observation. Let $d = \prod_{r \in R_P: \text{head}(r) = [p; \frac{1}{t}]} t$. Let $w \in \mathcal{W}$.

Recall that $P(w)$ is a finite product of the probabilities of a selection of rules from $R_P$. Consequently, $d \cdot P(w)$ is an integer. We will use this latter fact later in the proof.

We first prove that P-VER$_{sem}$ is in FP$^{#PH}$. To this end, we will specify a non-deterministic polynomial-time Turing Machine $M_1$ which will check, using a PH-oracle, if a given set $S \subseteq \mathcal{A}$ is an extension of $\mathcal{F}$ under a given semantics, and will compute the probability $P(w)$ of the possible world $w$. It will then split the computation tree into $d \cdot P(w)$ accepting branches. We will then use another Turing Machine, $M_2$, to count the accepting paths of $M_1$, which will place $M_2$ in PH. The output of $M_2$ will simply be the probability of $S$ being an extension, multiplied by $d$, so that P-VER$_{sem}$ will fall in FP$^{#PH}$.

First, we specify $M_1$ as follows:

a) Non-deterministically guess a possible world $w$. (In P.)

b) Check if $S$ is a $sem$ extension of $\mathcal{F}$, and continue only if so, returning ‘No’ otherwise. (In PH, by results in Section 4.)
c) Compute $P(w)$. (In PH, by Lemma 6.1.)

d) Split the computation tree $d \cdot P(w)$ times, $d \cdot P(w)$ being an integer (as proved earlier). Return ‘Yes’ in each of these $d \cdot P(w)$ branches, making them accepting branches. (In P.)

So $M_1$ is a non-deterministic polynomial-time Turing machine with a PH oracle. Specifically, if the input $S \subseteq A$ is a sem extension of $F_w$, then it returns ‘Yes’ in $d \cdot P(w)$ branches.

Now simply let $M_2$ to count the number of accepting paths of $M_1$. Thus, $M_2$ returns $d \cdot P(w)$ if $S$ is a sem extension of $F_w$. Clearly, $M_2$ is in $\#P^{PH} = \#PH$ (see Section 2.3).

Then our overall algorithm is the following:

1. Compute $d$;
2. Given $S \subseteq A$, call $M_2$, returning integer $N(S)$;
3. Return $\frac{N(S)}{d}$.

Indeed, let $N(S)$ be the output of $M_2$, i.e. the number of accepting paths of $M_1$ on input $S$. Then, by construction of $M_1$:

$$N(S) = \sum_{w \in W'} d \cdot P(w) = d \cdot \text{P-VER}_{sem}(S).$$

Thus, $\frac{N(S)}{d} = \text{P-VER}_{sem}(S)$. So this algorithm returns P-$\text{VER}_{sem}(S)$ with $S \subseteq A$ as input.

Computing $d$ is polynomial with $O(|R_P|)$ operations. Since $M_2$ counts the number of accepting paths of $M_1$ and is in #PH, we have that P-$\text{VER}_{sem} \in FP^{#PH}$.

Finally, as $FP^{#PH} = FP^{#P}$ (see Section 2.3), we have shown that P-$\text{VER}_{sem}$ is in $FP^{#P}$ for any sem $\in \{adm, prf, stb, set-stb, cpl, w-f\}$, as required. □

For lower bounds, we instantiate $F_P$ with a Horn ABA framework to obtain a PABA instance, called Horn-PABA henceforth. We will establish that for Horn-PABA probabilistic set verification is $FP^{#P}$-c (for Turing reduction). To do that we need the following result which says that for Turing reduction, #P problems are as hard as $FP^{#P}$ problems.

**Lemma 6.3.** A problem #P-hard for Turing reduction is also $FP^{#P}$-hard for Turing reduction.

**Proof.** Let $L_1$ be #P-hard and let $L_2$ be in $FP^{#P}$. We need to prove that $L_2 \leq_T L_1$. As $L_2 \in FP^{#P}$, there is a deterministic polynomial-time Turing machine $M_2$ and a #P problem $L_3$ such that $M_2$ solves $L_2$ using a polynomial number of calls to an oracle for $L_3$. As $L_1$ is #P-hard, $L_3 \leq_T L_1$. Hence, $L_3 \in FP^{L_1}$, and there is a deterministic polynomial-time Turing machine $M_3$ such that $M_3$ solves $L_3$ using a polynomial number of calls to an oracle for $L_1$. Consider the deterministic Turing machine $M$ which is a combination of $M_2$ and $M_3$ with an oracle for $L_1$ and solves $L_2$. As both $M_2$ and $M_3$ are deterministic polynomial-time and call their respective oracles for $L_3$ and $L_1$ a polynomial number of times, $M$ is a deterministic polynomial-time Turing machine solving $L_2$ using a polynomial number of calls to an oracle for $L_1$ oracle. Thus, $L_2 \leq_T L_1$. Consequently, $L_1$ is $FP^{#P}$-hard for Turing reduction. □
We are now ready to establish the lower bounds for probabilistic verification in Horn-PABA.

**Theorem 6.4 (P-VER Lower Bounds).** For a (flat or generic) Horn-PABA framework \( \mathcal{F}_p \) and \( \text{sem} \in \{ \text{adm}, \text{prf}, \text{stb}, \text{set-stb}, \text{cpl}, \text{w-f} \} \), \( \text{P-VER}_{\text{sem}} \) is \( \text{FP}^{\#P} \cdot \text{c} \) (for Turing reduction).

**Proof.** First, we prove \( \text{P-VER}_{\text{sem}} \) is \( \text{FP}^{\#P} \)-hard (for Turing reduction) by reducing a \( \text{FP}^{\#P} \)-problem \( \text{P2DNF} \) [42] to \( \text{P-VER}_{\text{sem}} \), for any \( \text{sem} \). \( \text{P2DNF} \) is the problem of counting the number of satisfying assignments of a disjunctive normal form (DNF) formula \( \Phi = C_1 \lor \ldots \lor C_k \) over a set \( X = \{ x_1, \ldots, x_n \} \) of positive variables, where for each \( i \in \{1, \ldots, k\} \) there are \( f, l \in \{1, \ldots, n\} \) such that \( C_i = x_j \land x_l \).

Given an instance \( \Phi \) of \( \text{P2DNF} \), define a PABA framework \( \mathcal{F}^\Phi = (\mathcal{A}^\Phi, \mathcal{R}^\Phi, \mathcal{F}_p) \) thus:

- \( \mathcal{A}^\Phi_p = X = \{ x_1, \ldots, x_n \}; \)
- \( \mathcal{R}^\Phi_p = \{ \{ i \} : i \in \{1, \ldots, n\} \}; \)
- \( \mathcal{F}_p = (\mathcal{L}^\Phi_p, \mathcal{R}^\Phi_p, \mathcal{F}_p, \emptyset) \) with
  - \( \mathcal{A}_p = \{ \} \),
  - \( \mathcal{R}_p = \{ \{ i \} : i \in \{1, \ldots, n\} \} \cup \{ \{ x_j, x_l \} : i \in \{1, \ldots, k\}, C_i = x_j \land x_l \} \).

Note that the underlying ABA framework \( \mathcal{F}_p \) of \( \mathcal{F}^\Phi \) is flat. Showing \( \text{FP}^{\#P} \)-hardness (for Turing reduction) of probabilistic verification for flat Horn-PABA will also show it for generic (possibly non-flat) Horn-PABA. Also note that by construction of \( \mathcal{F}_p \), its extensions coincide under all semantics, in either flat or non-flat case; in particular, an admissible extension of \( \mathcal{F}_p \) is grounded (well-founded), complete, preferred, stable and set-stable. Thus, we can fix \( \text{sem} = \text{adm} \).

To facilitate the proof, we represent an assignment of \( \Phi \) as a \( \subseteq \)-maximally consistent subset of \( \mathcal{A}^\Phi_p \cup \neg \mathcal{A}^\Phi_p \), i.e. a possible world. Consider an assignment \( w \). If it makes \( \Phi \) true, then by construction of \( \mathcal{R}_p \), \( \emptyset \models \neg \mathcal{F}_w \) is an argument in the associated ABA framework \( \mathcal{F}^\Phi_w \) (extensions of which coincide under all semantics, just as in \( \mathcal{F}_p \)). Thus, \( \{ \} \) is not admissible in \( \mathcal{F}^\Phi_w \). Conversely, if \( \{ \} \) is not admissible in \( \mathcal{F}^\Phi_w \), then \( \emptyset \models \mathcal{F}_w \) is an argument in \( \mathcal{F}^\Phi_w \). Hence, there is \( i \in \{1, \ldots, k\} \) such that \( \emptyset \models C_i \) is an argument in \( \mathcal{F}^\Phi_w \), which means that \( \emptyset \models x_j \) and \( \emptyset \models x_l \) are arguments in \( \mathcal{F}_w \) too, where \( C_i = x_j \land x_l \). Then \( x_j, x_l \in w \), and so \( w \) makes \( \Phi \) true. Consequently, \( w \) makes \( \Phi \) true iff \( \{ \} \) is not a \( \text{sem} \) extension of \( \mathcal{F}^\Phi_w \).

Now let \( N \) be the number of satisfying assignments of \( \Phi \). Then \( N \) is the number of possible worlds for which \( \{ \} \) is not a \( \text{sem} \) extension of \( \mathcal{F}^\Phi_w \), where \( w \in \mathcal{W} \). Note that by construction of \( \mathcal{R}^\Phi_p \), there are \( 2^n \) possible worlds, each having the same probability.

---

9As stated in [42], even the very restricted problem of counting the number of satisfying assignments of a monotone CNF formula (over a set of only positive or only negative variables) with two disjuncts per conjunct is \( \text{FP}^{\#P} \cdot \text{c} \). However, the same applies to counting satisfying assignments of (monotone) DNF formulas with two conjuncts per disjunct. Indeed, the satisfying assignments of \( \Phi \) (in DNF) are falsifying assignments of \( \neg \Phi \) (which is in CNF). So counting for \( \Phi \) and subtracting the result from the number of all possible assignments (i.e. from \( 2^n \)) yields the number of satisfying assignments of \( \neg \Phi \). I.e., counting for CNF or DNF reduces to each other.
1/n. So the probability for \( \{f\} \) not to be a sem extension of \( \mathcal{F}_w^\Phi \) for some \( w \in \mathcal{W} \) is \( \frac{1}{2^n} \). Therefore, \( N = 2^n(1 - \text{P-VER}_{\text{sem}}(\{f\})) \) can be computed in polynomial time by using an oracle for the probabilistic verification problem P-VER_{sem} in \( \mathcal{F}_w^\Phi \).

We have just shown that \#P2DNF is Turing reducible to P-VER_{sem}. So P-VER_{sem} is \#P-hard (for Turing reduction). Thus, according to Lemma 6.3, P-VER_{sem} is FP^{#P}-hard (for Turing reduction). Using Theorem 6.2 we conclude that P-VER_{sem} is FP^{#P}-c (for Turing reduction).

\[ \square \]

6.2. Probabilistic Sceptical and Credulous Acceptance in PABA

We now show that P-CA and P-SA for PABA and Horn-PABA are, respectively, in and complete for the same class FP^{#P} as P-VER. We establish the upper bound by a natural variation of the algorithm used for P-VER, and the lower bound by noting that CA and SA coincide with VER when mapping \#P2DNF to Horn-PABA.

**Theorem 6.5 (P-CA and P-SA Upper Bounds).** For a generic PABA framework \( \mathcal{F}_p \) and \( \text{sem} \in \{adm, prf, stb, set-stb, cpl, w-f\} \), P-CA_{sem} and P-SA_{sem} are in FP^{#P}.

**Proof.** Analogously to Theorem 6.2, with the following change. In step b) of the specification of \( M_1 \), instead of checking if \( S \) is a sem extension, \( M_1 \) will check if \( \pi \in \mathcal{L} \) is accepted credulously or sceptically under sem semantics. This is also done by a PH-oracle, according to results in Section 4.

\[ \square \]

**Theorem 6.6 (P-CA and P-SA Lower Bounds).** For a (flat or generic) Horn-PABA framework \( \mathcal{F}_p \) and \( \text{sem} \in \{adm, prf, stb, set-stb, cpl, w-f\} \), P-CA_{sem} and P-SA_{sem} are FP^{#P}-c (for Turing reduction).

**Proof.** We show that P-CA_{sem} and P-SA_{sem} are \#P-hard (for Turing reduction) by reducing \#P2DNF. As in the proof of Theorem 6.4, for an instance \( \Phi \) of \#P2DNF and the flat PABA framework \( \mathcal{F}_w^\Phi \), note that for each possible world \( w \in \mathcal{W} \), the associated ABA framework \( \mathcal{F}_w^\Phi \) has one and only one extension under any semantics: either \( \emptyset \) or \( \{f\} \). Hence, for any \( \text{sem} \in \{adm, prf, stb, set-stb, cpl, w-f\} \), \( \{f\} \) is a sem extension of \( \mathcal{F}_w^\Phi \) iff \( f \) is credulously accepted in \( \mathcal{F}_w^\Phi \) under sem semantics iff \( f \) is sceptically accepted in \( \mathcal{F}_w^\Phi \) under sem semantics. Thus, P-VER_{sem}(\{f\}) = P-CA_{sem}(f) = P-SA_{sem}(f).

So \#P2DNF is Turing reducible to both P-CA_{sem} and P-SA_{sem}, and so P-CA_{sem} and P-SA_{sem} are \#P-hard (for Turing reduction). From Lemma 6.3 and Theorem 6.5 it follows that they are FP^{#P}-c (for Turing reduction).

\[ \square \]

Table 10 (shortened version of a part (the left side) of Table 4 in Section 1) summarises the complexity results for function problems in PABA as proved in this section.

<table>
<thead>
<tr>
<th>sem</th>
<th>PABA (flat and generic)</th>
<th>Horn-PABA (flat and generic)</th>
</tr>
</thead>
<tbody>
<tr>
<td>adm, cpl, prf, stb, set-stb, w-f/gld</td>
<td>FP^{#P}</td>
<td>FP^{#P}-c</td>
</tr>
</tbody>
</table>

Table 10: Summary of new complexity results for function problems in PABA. The upper bounds (column 2) are general for PABA, assuming the derivation problem in ABA is within PH. The lower bounds (column 3) are specific for (either flat or generic) Horn-PABA. The same upper and lower complexity bounds, respectively, apply to all three complexity problems for PABA, under all semantics in consideration.
From the outset, determining the probability with which a given sentence is acceptable in PABA requires going over all the possible worlds, which is exponential in the number of probabilistic assumptions. This is also reflected in (the proofs of) the results obtained in this section indicating that counting the number of accepting paths of a non-deterministic polynomial-time Turing machine is required. Our complexity results suggest that in order to efficiently solve the function problems delineated in Section 5.3 one could try to exploit randomised (also called probabilistic) algorithms that are often used to find approximate solutions to \( \text{FP}^{\text{NP}} \cdot \text{c} \) problems.

The only existing implementation of PABA (to the best of our knowledge), presented by Hung in [30], is, however, non-randomised. Instead, it exploits the idea of looking for partial possible worlds that contain few probabilistic assumptions but still make a sentence acceptable, and then determining the probability of the actual possible world being consistent with the partial ones. This seems to be specifically of use for the so-called Bayesian PABA frameworks, in which the probabilistic information corresponds exactly to a Bayesian network, whereby efficient algorithms for Bayesian inference can be employed. In general, though, this method would not help to efficiently deal with non-Bayesian PABA frameworks due to the exponential blow-up in going through all the possible worlds to determine probabilistic acceptance of a sentence. Instead, one could think of restricted reasoning problems where, for instance, one wants to determine whether a given sentence is acceptable in some worlds ‘that matter’. We propose two specific, novel (decision) problems of this kind in the next section and investigate their complexity.

6.2.1. Strong Complexity Problems in PABA

We here propose and study new complexity problems in PABA. Specifically, we propose decision problems for CA and SA with the given probability of 1. In other words the following strong acceptance problems concern deciding whether a given sentence is credulously/sceptically accepted almost everywhere.

**Definition 6.1.** Let \( \mathcal{F}_P = (\mathcal{A}_P, \mathcal{R}_P, \mathcal{F}) \) be a PABA framework with the underlying ABA framework \( \mathcal{F} = (\mathcal{L}, \mathcal{R}, \mathcal{A}, \mathcal{F}) \) and \( \text{sem} \in \{\text{adm}, \text{prf}, \text{stb}, \text{set-stb}, \text{cpl}, \text{w-f}\} \).

- The **strong credulous acceptance** problem for a given \( \pi \in \mathcal{L} \), \( \text{StrongCA}_{\text{sem}}(\pi) \), is the problem of deciding whether \( \pi \) is credulously accepted under \( \text{sem} \) semantics with probability 1.
  I.e. \( \text{StrongCA}_{\text{sem}}(\pi) \) amounts to deciding whether
  \[
  \sum_{w \in \mathcal{W}} \sum_{\exists S \subseteq \mathcal{A} : \mathcal{F}_w \vdash_{\text{sem}} S, \pi \in \text{Cn}(S)} P(w) = 1.
  \]

- The **strong sceptical acceptance** problem for a given \( \pi \in \mathcal{L} \), \( \text{StrongSA}_{\text{sem}}(\pi) \), is the problem of deciding whether \( \pi \) is sceptically accepted under \( \text{sem} \) semantics with probability 1.
  I.e. \( \text{StrongSA}_{\text{sem}}(\pi) \) amounts to deciding whether
  \[
  \sum_{w \in \mathcal{W}} \sum_{\forall S \subseteq \mathcal{A} : \mathcal{F}_w \vdash_{\text{sem}} S \rightarrow \pi \in \text{Cn}(S)} P(w) = 1.
  \]
These problems are not function problems as in Section 5.3, but decision problems as in Section 3.4. They can therefore be solved with oracles for, respectively, credulous and sceptical acceptance in ABA, as we show next.

**Theorem 6.7 (StrongCA and StrongSA Upper Bounds).** For a generic PABA framework \( \mathcal{F}_P \) and \( \text{sem} \in \{ \text{adm}, \text{prf}, \text{stb}, \text{set-stb}, \text{cpl}, \text{w-f} \} \), \( \text{StrongCA}_{\text{sem}} \) and \( \text{StrongSA}_{\text{sem}} \) are in \( \text{coNP}^C \) (for PABA), where \( \text{CA}_{\text{sem}} \) and \( \text{SA}_{\text{sem}} \), respectively, are in \( C \) (for ABA).

**Proof.** Suppose \( \text{CA}_{\text{sem}} \) (respectively, \( \text{SA}_{\text{sem}} \)) can be solved by a \( C \)-oracle \( M \). Then the following solves the complement of \( \text{StrongCA}_{\text{sem}}(\pi) \):

- Guess a possible world \( w \);
- Check that \( \pi \in \mathcal{L} \) is not credulously accepted in \( F_w \) under \( \text{sem} \) semantics using \( M \).

This ensures that the complement of \( \text{StrongCA}_{\text{sem}} \) is in \( \text{NP}^C \). Thus, \( \text{StrongCA}_{\text{sem}} \) is in \( \text{coNP}^C \).

An analogous algorithm works for the complement of \( \text{StrongSA}_{\text{sem}}(\pi) \), checking instead if \( \pi \) is not sceptically accepted. Hence, \( \text{StrongSA}_{\text{sem}} \) is in \( \text{coNP}^C \) too.

Given the upper bounds for credulous and sceptical acceptance problems under various semantics summarised in Table 7 in Section 4, we deduce and summarise in Table 11 (same as a part (the right side) of Table 4 in Section 1) results for strong complexity problems in PABA.

<table>
<thead>
<tr>
<th>( \text{sem} )</th>
<th>Flat PABA</th>
<th>Generic PABA</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{adm} )</td>
<td>(( \Pi^P ))(^D)</td>
<td>(( \Pi^P ))(^D)</td>
</tr>
<tr>
<td>( \text{cpl} )</td>
<td>(( \Pi^P ))(^D)</td>
<td>(( \Pi^P ))(^D)</td>
</tr>
<tr>
<td>( \text{prf} )</td>
<td>(( \Pi^P ))(^D)</td>
<td>(( \Pi^P ))(^D)</td>
</tr>
<tr>
<td>( \text{stb} )</td>
<td>(( \Pi^P ))(^D)</td>
<td>(( \Pi^P ))(^D)</td>
</tr>
<tr>
<td>( \text{set-stb} )</td>
<td>(( \Pi^P ))(^D)</td>
<td>(( \Pi^P ))(^D)</td>
</tr>
<tr>
<td>( \text{grd/w-f} )</td>
<td>( \text{coNP}^D )</td>
<td>( \text{coNP}^D )</td>
</tr>
</tbody>
</table>

Table 11: Summary of new upper bound results for strong complexity problems in PABA. \( D \) is the complexity class of the derivation problem in ABA.

It would undoubtedly be interesting to study if these upper bounds on the complexity of the two problems above are tight by studying the lower bounds of, for example, the Horn-PABA instance. We leave this for future work.

### 7. Related Works

Probabilistic argumentation can be classified [16, 20, 21] into two types of approaches: the constellations approach, e.g. [18, 22, 23, 24, 25, 26, 27]; and the epistemic approach, e.g. [28, 20]. Formalisms within the epistemic approach assume a fixed topology of an (abstract) argumentation graph, with given probabilities over arguments and attacks representing the degree of belief in them. The reasoning then amounts to establishing the probabilistic acceptance of arguments given the relationship among them.
expressed via attacks, where the main idea is that the more an argument is believed in, the less its attackers should be believed in. This approach consequently requires the probability distribution to be compatible with the given probabilities over arguments and attacks, and in principle necessitates considerations of all possible probability distributions over the sets of possible worlds. The upshot is that one aims to find ‘epistemic extensions’ which amount to sets of arguments believed in to some extent.

Instead, formalisms within the constellations approach assume an uncertain topology of an argumentation graph (an abstract one or one spawned by a structured argumentation framework, e.g. as in PABA), and then focus on probability assignments to arguments, attacks and argumentation frameworks themselves. The reasoning there amounts to establishing the probabilistic acceptance of arguments and sentences using standard argumentation semantics of extensions, given the uncertainty of the argumentation graph/framework as expressed by a probability distribution over the graphs or possible worlds. Importantly, these formalisms can be further distinguished by their use of the so-called independence assumption, whereby arguments are assumed to be pairwise independent and attacks are dependent only on the arguments they relate but not on other attacks [27, 21]. Under the independence assumption, the probabilities of the argumentation frameworks can be computed from the given probabilities over arguments and attacks. For instance, from the above mentioned approaches, [22, 26, 27] make the independence assumption. Instead, PABA and the other probabilistic structured argumentation (PSA) formalism p-ASPIC [24] impose no independence restrictions on the probability distribution on (the probabilistic assumptions and rules generating) the arguments. We also note that PABA is expressive enough to capture not only p-ASPIC, but also the independence-assuming approach of [22], essentially because the semantics of those approaches can be seen as probabilistic versions of classical AA semantics [30]. For a more detailed discussion on the differences among various probabilistic argumentation formalisms we refer the reader to [16, 30, 21].

The complexity of formalisms following the constellations approach has been most prominently studied in [27, 35, 21]. In particular, Probabilistic Abstract Argumentation (PAA) of the kind considered in [22] was addressed by Fazzinga et al. in [27]. Further, Fazzinga et al. addressed Probabilistic Bipolar Argumentation (PBA) under deductive support [55], with and without the independence assumption, in [35]. Specifically, the authors of these two papers studied the probabilistic set verification problem P-VER under admissible, complete, preferred, grounded, stable and ideal [56, 34] semantics. With respect to the variants of PAA and PBA they considered, they established that P-VER ranges over classes \(FP, FP^{NP}\) (the class of function problems solvable by a deterministic polynomial-time Turing machine with parallel calls to an NP-oracle) and \(FP^{#P}\), also being complete for them in many cases. Fazzinga et al. further extended these results in [21] to cover the probabilistic credulous acceptance problem P-CA for various forms of PAA with similar results (but with different classes under semi-stable [37] semantics). The authors also noted, but left as future work, that the analysis of P-CA can be adapted for the analysis of the probabilistic sceptical acceptance problem P-SA in PAA. To the best of our knowledge, the complexity of (any variant of) PSA has not until now been studied. This paper contributes such a study.

Complexity of structured argumentation (see e.g. [3, Part II] and [57] for overviews) has been investigated (see [39] for a recent overview of complexity studies in compu-
tational argumentation), but may still be viewed as “seriously underdeveloped” [58].
In this paper we aimed to address this issue by considerably expanding the details on
the complexity of ABA. Previous work on the latter includes, most notably, [32] as
discussed in Section 3.1, and Dunne’s work on ideal semantics in [34], as well as the
recent works on the complexity of flat ABA with preferences (ABA+ [59, 60, 40])
[61] and of Bipolar ABA ([36], a form of non-flat ABA) [47]. With respect to [32], we
complemented the existing results with new results under a larger set of semantics (aug-
menting with complete, stable, set-stable and well-founded semantics), for additional
problems (particularly existence \(EX\)) and in a new ABA instance (Horn-ABA). Consol-
idating our results with [34] yields a largely comprehensive picture of the complexity
of flat and generic ABA (without probabilities).

This picture is highly relevant with respect to other structured argumentation for-
malisms. In particular, ASPIC+ [62, 31, 58] without preferences (also known simply as
ASPIc) can be captured in flat ABA [63] (and vice versa [58]), so that our complexity
results may apply to ASPIC too. Similarly, according to Borg and Straßer in [64], flat
ABA can also be captured in Sequent-Based Logical Argumentation [65, 66, 67], so
that our complexity results may apply there too. Still further, some other formalisms,
such as Deductive Argumentation [68, 69, 70] and approaches [71, 72], can be cap-
tured in ASPIC+, so that our results may be informative as regards the complexity of
those formalisms, provided preferences are taken into account. Any such analysis is
beyond the scope of this paper but may provide interesting venues for future research
in complexity of structured argumentation.

It is worth noting that structured argumentation formalisms such as ASPIC+, ASPIC-
like approaches [73, 74], Deductive Argumentation and Sequent-Based Logical Argumentation operate by first constructing the arguments and attacks among them, and then
using AA semantics for determining extensions. Flat ABA can also operate equival-
ently via this two-step process [56, 75, 40] (though generic ABA has not yet been char-
acterised in such terms). The complexity of reasoning in such a setting then amounts to
the cumulative complexity of (i) an AA framework construction based on the underly-
ing logic, (ii) determining extensions under various semantics in AA, and (iii) extract-
ing conclusions from the acceptable arguments. Task (ii) is well-studied (see e.g. [39]
for an overview), whereas the other two tasks are less so. Notable exceptions include
[76, 77, 78, 79] and the work on the complexity of ABA (including also [80]), to which
this article contributes substantially by exploring the complexity of reasoning directly
from the underlying logic to acceptable conclusions. Another notable structured argu-
mentation formalism employing such a direct approach (with semantics differing from
those of AA) is Defeasible Logic Programming (DeLP) [81, 82], with its early variants
having some complexity results established in [83]. The latter refer to data complex-
ity of the DeLP game-based semantics and therefore are not directly comparable to
complexity results in this paper. Nonetheless, conceptually the complexity problems in
[83] can be considered akin to credulous acceptance under admissible semantics in flat
ABA, and they are accordingly in \(NP\) (albeit in contrast to the derivation oracle for
ABA, they require a different kind of C oracle that answers “whether an argument can
be considered in the tree structure of a game” [83]). Technical analysis of complexity
connections between DeLP and ABA is beyond the scope of this paper but may prove
interesting in the future.
Finally, we would like to mention a few related works in terms of Logic Programming (LP). In [50] the authors studied the relationship between flat ABA and LP. Specifically, Caminada and Schulz implicitly considered what we deemed in this paper flat Horn-ABA and established its correspondence with normal logic programs under various (ABA and LP) semantics. Since LP-ABA can be seen as flat Horn-ABA, our new complexity results for LP-ABA may be transferred to LP and give rise to complexity results under various LP semantics, where applicable. While this is beyond the scope of the current paper, it may be an interesting future work direction.

We also note that Horn-ABA can be seen as an instance of ABA based on a fragment of the general extended disjunctive programs (GEDPs) [51] (with strong negation, disjunctions and negation as failure in the heads), namely the fragment amounting to logic programs with negation as failure but not disjunction in the heads. Reasoning with GEDPs at large under answer set semantics makes existence, credulous acceptance and sceptical acceptance problems $\Sigma^p_2$-$c$, $\Sigma^p_2$-$c$ and $\Pi^p_2$-$c$, respectively [51, Theorem 6.4]. Our results on the complexity of Horn-ABA under stable semantics can potentially establish that reasoning with the fragment in question is easier ($\mathcal{NP}$-$c$, $\mathcal{NPC}$ and $\mathcal{coNP}$-$c$, respectively), provided that there is a one-to-one correspondence between Horn-ABA under stable semantics and this fragment under answer set semantics. We conjecture that it is indeed so, because [29, Theorem 3.10] can be easily adapted to Horn-ABA and the fragment of GEDPs without disjunctions. We leave verification of this conjecture as well as the study of other relevant issues pertaining to LP for future work.

8. Conclusions

Our goal was to establish the computational complexity of PABA—Probabilistic Assumption-Based Argumentation [18, 30], a variant of probabilistic structured argumentation (PSA). To this end, as necessary for studying the complexity of PABA, we first analysed the complexity of (non-probabilistic) ABA, in general as well as for its instances LP-ABA and Horn-ABA for flat and generic cases, respectively. Specifically, we investigated the existence problem $\mathcal{EX}$, consequently revisited the sceptical acceptance problem $\mathcal{SA}$, and thence complemented the existing ABA complexity results with new results for these two as well as the credulous acceptance $\mathcal{CA}$ and set verification $\mathcal{VER}$ problems, under admissible, complete, preferred, stable, set-stable and well-founded/grounded semantics. Our results concerning ABA (without probabilities), summarised in Tables 7, 8 and 9, provide a comprehensive picture of its complexity landscape, unprecedented and highly relevant in terms of other structured argumentation formalisms.

As regards PABA, we analysed the complexity problems $\mathcal{P}$-$\mathcal{VER}$, $\mathcal{P}$-$\mathcal{CA}$ and $\mathcal{P}$-$\mathcal{SA}$ under all the semantics considered in this paper, in cases of either flat or generic PABA frameworks and their instantiations. We established that, in terms of general upper bounds, under all semantics these problems lie in the class $\mathcal{FP}^{\mathcal{HP}}$ of function problems with a $\#\mathcal{P}$ oracle for counting the solutions of an $\mathcal{NP}$ problem, for PABA frameworks with the derivation problem within the Polynomial Hierarchy $\mathcal{PH}$. In terms of lower bounds, we established $\mathcal{FP}^{\mathcal{HP}}$-completeness for a newly defined PABA instance Horn-PABA of PABA frameworks instantiated with Horn-ABA. We further proposed and analysed the complexity of novel decision problems concerning (strong) acceptance
with probability 1 in PABA, establishing their upper bounds in cases of both flat and
generic PABA frameworks. Our investigations of PABA, with results summarised in
Tables 10 and 11, provide solid theoretical foundations to the complexity studies of
ABA with probabilities, and open new research directions within PSA in particular and
probabilistic argumentation at large.

Regarding computational considerations, our results showing that reasoning in PABA
requires counting with a $\#P$ oracle suggest that randomised algorithms could be used
for approximating probabilistic verification/acceptance in PABA, as opposed to the ex-
isting method that exhaustively explores the space of all possible worlds [30]. It is rea-
sonable to expect that the same would apply to implementation considerations in PSA
at large. On the other hand, our newly defined strong acceptance (decision) problems
in PABA and their complexity analysis hint at the possibility of developing algorithms
that concern acceptability in ‘the worlds that matter’ as given by some threshold proba-
bility, e.g. ‘almost everywhere’ (probability 1). Similar considerations may well apply
to other probabilistic argumentation approaches, especially those captured by PABA.
Our analysis therefore opens up new and potentially fruitful avenues of research.

Concerning future work, in addition to that already mentioned in Section 7 as well
as this section, in terms of PABA it would interesting to analyse if the upper bounds
on the complexity of strong credulous and sceptical acceptance are tight by studying
the lower bounds of e.g. the Horn-PABA instance. Similarly, it would be interesting to
investigate parametrised versions of the strong acceptance problems, whereby the given
probability is a parameter rather than fixed to 1, or otherwise finding the argument(s)
with the highest/lowest probability of acceptance (as hinted e.g. in [21]). In terms of
ABA, we left open the question of tightness of the upper bounds for the Horn-ABA
instance under well-founded semantics with respect to $\text{VER}$, $\text{CA}$ and $\text{SA}$. It would also
be interesting to study complexity of some other semantics defined for ABA, such as
semi-stable extensions.

Acknowledgements

Kristijonas Čyra and Francesca Toni were supported by the EPSRC project (grant
number EP/P029558/1) ROAD2H: Resource Optimisation, Argumentation, Decision
Support and Knowledge Transfer to Create Value via Learning Health Systems.

The authors are extremely grateful to the anonymous reviewers for very helpful
feedback and especially for pointing out some issues in statements and proofs in the
first draft of the paper.

Data access statement. No new data was collected in the course of this research.

References

[1] S. Parsons, C. Sierra, N. Jennings, Agents that Reason and Negotiate by Arguing,


