Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200521

Consistency in Assumption-Based Argumentation

Toshiko WAKAKI [1]
Shibaura Institute of Technology, Japan

Abstract. There exist counterexamples to Schulz and Toni's theorems which are the basis of their approach for justifying answer sets using assumption-based argumentation (ABA) whose language contains explicit negation. Against their claims, we present theorems showing the correspondence between answer sets of a consistent extended logic program and consistent stable extensions of the ABA instantiated with it. We show such ABA is not ensured to satisfy the consistency postulates. We also propose the novel notion of consistency for admissible dispute trees to avoid consistency problems in ABA applications containing explicit negation. We show the condition under which ABA consistency is ensured.

Keywords. consistency postulates, ABA consistency, consistent extensions, consistent sets of assumptions, consistent admissible dispute trees

1. Introduction

Consistency in assumption-based argumentation (ABA, for short) [1] is crucial to avoid anomalies in ABA applications whose languages contain explicit negation. In ASPIC$^+$ [9], for example, as the ways in which arguments can be in conflict, it allows the rebutting attack between two arguments having the mutually contradictory conclusions w.r.t. explicit negation along with undercutting and undermining attacks, while some conditions (e.g. ensuring closure under *transposition* or *contraposition*) under which ASPIC$^+$ satisfies *rationality postulates* [2] have been proposed to avoid anomalous results. In contrast, ABA allows only attacks against the support of arguments as defined in terms of a notion of *contrary*, while the *ab-self-contradiction axiom* [7] was proposed as the condition to guarantee the *consistency property* in an ABA framework containing explicit negation.

Recently as one of ABA applications containing explicit negation, extended abduction in ABA [14] was presented on the basis of the newly proposed definition of *consistency* in a flat ABA framework, which is slightly different from the notion of satisfying the *consistency property* [7] or the *direct consistency postulate* [2].

On the other hand, as another ABA application, Schulz and Toni proposed the approach of justifying answer sets using argumentation [11], where they used flat ABA frameworks instantiated with consistent extended logic programs (ELPs, for short) containing classical negation [8], i.e. explicit negation. However they took account of neither rationality postulates in such ABAs nor inconsistent extensions. As a result, it reveals the

[1]Corresponding Author: Toshiko Wakaki; E-mail: twakaki@shibaura-it.ac.jp

serious problem that there exist counterexamples to their theorems [11, Theorems 1, 2] such that answer sets of a *consistent* ELP are captured by stable extensions of the ABA framework instantiated with the ELP, though they are the basis of their approach for justifying why a literal belongs to an answer set of an ELP based on ABA. Besides regarding their computational machinery, we found that according to their lemma [11, Lemma 11], there may arise admissible abstract dispute trees whose defence sets are *inconsistent* though they are *admissible*. To the best of our knowledge, the notion of consistency for the defence set of an admissible dispute tree [5] has not been taken into account so far.

In this paper, first we discuss consistent extensions and ABA consistency. Second we show counterexamples to Schulz and Toni's theorems [11, Theorems 1, 2]. Then against their claims, we present the theorems showing that there is a one-to-one correspondence between answer sets of a *consistent* ELP and (*not* stable extensions but) *consistent* stable extensions of the ABA instantiated with the ELP. Besides we show that such ABA instantiated with a consistent ELP is not ensured to satisfy the consistency property, which implies that their theorems are incorrect. Third as another serious problem, we show the admissible abstract dispute tree whose defence set is *inconsistent* though it is *admissible* as derived due to their lemma [11, Lemma 11]. Then to detect and avoid such anomaly, we propose the novel notion of *consistency* for admissible dispute trees. So far a simplified assumption-based framework [5] (a simplified ABA, for short) having the restricted form w.r.t. explicit negation has often been used to illustrate admissible dispute trees without addressing consistency. Instead, thanks to our notion of consistency, we can show that the serious consistency problems addressed above never occurs in a simplified ABA since any defence set of its admissible dispute tree is consistent. Finally we present the condition to ensure ABA consistency in comparison with the *ab-self-contradiction axiom* to guarantee consistency-property in ABA.

The rest of this paper is as follows. Section 2 shows preliminaries. Section 3 discusses consistency in ABA. Section 4 shows counter examples to their theorems, presents the corrected theorems, proposes (in)consistent admissible dispute trees and shows the condition to ensure ABA consistency. Section 5 discusses related work and concludes.

2. Preliminaries

Definition 1 An ABA framework (or ABA) [6,1] is a tuple $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$, where $(\mathcal{L}, \mathcal{R})$ is a deductive system, consisting of a language $\mathcal{L}$ (a set of sentences) and a set $\mathcal{R}$ of inference rules of the form: $b_0 \leftarrow b_1, \ldots, b_m$ ($b_i \in \mathcal{L}$ for $0 \leq i \leq m$), $\mathcal{A} \subseteq \mathcal{L}$ is a set of *assumptions*, and $\overline{}$ is a total mapping from $\mathcal{A}$ into $\mathcal{L}$. $\overline{\alpha}$ is referred to as the *contrary* of $\alpha \in \mathcal{A}$. For a rule $r \in \mathcal{R}$ of the form $b_0 \leftarrow b_1, \ldots, b_m$, let the head be $head(r) = b_0$ (resp. the body $body(r) = \{b_1, \ldots, b_m\}$).

We enforce that ABA frameworks are *flat*, namely assumptions do not occur in the head of rules. In ABA, *arguments* and $attacks$ are defined as follows [6]:

- an *argument for* $c \in \mathcal{L}$ (*the conclusion* or *claim*) *supported by* $K \subseteq \mathcal{A}$ ($K \vdash c$ in short) is a (finite) tree with nodes labelled by sentences in $\mathcal{L}$ or by $\tau \notin \mathcal{L}$ denoting "true", such that the root is labelled by c, leaves are labelled either by τ or by assumptions in K, and each non-leaf node N is labelled by $b_0 = head(r)$ for some rule $r \in \mathcal{R}$, where N has a child labelled by τ if $body(r) = \emptyset$; otherwise N has m children, each of which is labelled by $b_j \in body(r) = \{b_1, \ldots, b_m\} (1 \leq j \leq m)$.

- *an argument* $K_1 \vdash c_1$ *attacks an argument* $K_2 \vdash c_2$ iff $c_1 = \overline{\alpha}$ for $\alpha \in K_2$ (1)

Let $AF_{\mathcal{F}} = (AR, attacks)$ be the abstract argumentation framework generated from a flat ABA framework $\mathcal{F}$. For $Args \subseteq AR$, let $Args^+ = \{B \in AR \mid A \; attacks \; B$ for $A \in Args\}$. $Args$ is *conflict-free* iff $Args \cap Args^+ = \emptyset$. $Args$ *defends* an argument A iff each argument that attacks A is attacked by an argument in $Args$.

Definition 2 ***(ABA semantics)*** [1,4] Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ be a flat ABA framework, and AR the associated set of arguments. Then $Args \subseteq AR$ is: admissible iff $Args$ is conflict-free and *defends* all its elements; a complete argument extension iff $Args$ is admissible and contains all arguments it *defends*; a preferred (resp. grounded) argument extension iff it is a (subset-)maximal (resp. (subset-)minimal) complete argument extension; a stable argument extension iff it is conflict-free and $Args \cup Args^+ = AR$; an ideal argument extension iff it is a (subset-)maximal admissible set contained in every preferred argument extension.

Hereafter let $\sigma \in \{$complete, preferred, grounded, stable, ideal$\}$. The various ABA semantics is originally given by sets of assumptions called assumption extensions. There is a one-to-one correspondence between σ assumption extensions and σ argument extensions such that for a σ assumption extension $Asms$, $\texttt{Asms2Args}(Asms) = \{K \vdash c \in AR \mid K \subseteq Asms\}$ is a σ argument extension, and for a σ argument extension $Args$, $\texttt{Args2Asms}(Args) = \{\alpha \in K \mid K \vdash c \in Args, K \subseteq \mathcal{A}\}$ is a σ assumption extension [3]. Let *claim*(Ag) be the claim (or conclusion) of an argument Ag. Then the *conclusion* of a set of arguments $\mathcal{E}$ is $\texttt{Concs}(\mathcal{E}) = \{c \in \mathcal{L} \mid c = claim(Ag)$ for $Ag \in \mathcal{E}\}$, while the *consequences* of a set of assumptions $A \subseteq \mathcal{A}$ is $Cn(A) = \{s \in \mathcal{L} \mid \exists A' \vdash s$ for $A' \subseteq A\}$.

Definition 3 ***(Dispute trees)*** [5] Given a flat ABA framework $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$, an abstract dispute tree for an initial argument a is defined as a (possibly infinite) tree $\mathcal{T}$ such that **1.** Every node of $\mathcal{T}$ is labelled by an argument and is assigned the status of *proponent* node or *opponent* node, but not both. **2.** The root is a proponent node labelled by a. **3.** For every proponent node N labelled by an argument b, and for every argument c that attacks b, there exists a child of N, which is an opponent node labelled by c. **4.** For every opponent node N labelled by an argument b, there exists exactly one child of N which is a proponent node labelled by an argument which attacks some assumption α in the set supporting b. α is said to be the *culprit* in b. **5.** There are no other nodes in $\mathcal{T}$ except those given by **1-4** above.

The set of all assumptions belonging to the proponent nodes in $\mathcal{T}$ is called the defence set of $\mathcal{T}$, denoted by $\mathcal{D}(\mathcal{T})$. An abstract dispute tree $\mathcal{T}$ is admissible if and only if no culprit in the argument of an opponent node belongs to $\mathcal{D}(\mathcal{T})$. If $\mathcal{T}$ is an admissible abstract dispute tree for an argument a, then $\mathcal{D}(\mathcal{T})$ is an admissible set of assumptions. If a is an argument supported by a set of assumptions $A_0 \subseteq E$ where E is admissible, then there exists an admissible dispute tree $\mathcal{T}$ for a with defence set $\mathcal{D}(\mathcal{T})$ and $A_0 \subseteq \mathcal{D}(\mathcal{T}) \subseteq E$ and $\mathcal{D}(\mathcal{T})$ is admissible [5, Theorem 5.1].

Satisfying Caminada and Amgoud's rationality postulates [2] or the *closure* and *consistency* properties in ABA [7] is stated as follows.

Definition 4 ***(Rationality postulates)*** [7,2] Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ be a flat ABA framework. A set $X \subseteq \mathcal{L}$ is said to be contradictory iff X is contradictory w.r.t. $\overline{}$, i.e. there exists an assumption $\alpha \in \mathcal{A}$ such that $\{\alpha, \overline{\alpha}\} \subseteq X$; or X is contradictory w.r.t. $\neg$, i.e. there exists $s \in \mathcal{L}$ such that $\{s, \neg s\} \subseteq X$ if $\mathcal{L}$ contains an explicit negation operator $\neg$. Let $CN_{\mathcal{R}} : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{L}}$ be a consequence operator. For a set $X \subseteq \mathcal{L}$, $CN_{\mathcal{R}}(X)$ is the smallest set such that $X \subseteq CN_{\mathcal{R}}(X)$, and for each rule $r \in \mathcal{R}$, if $body(r) \subseteq CN_{\mathcal{R}}(X)$ then $head(r) \in CN_{\mathcal{R}}(X)$. X is closed iff $X = CN_{\mathcal{R}}(X)$. A set $X \subseteq \mathcal{L}$ is said to be inconsistent iff its closure $CN_{\mathcal{R}}(X)$ is contradictory. X is said to be consistent iff it is not inconsistent. A flat ABA framework $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ \ } \rangle$ is said to satisfy the *consistency-property* (resp. the *closure-property*) if for each complete extension $\mathcal{E}$ of $AF_{\mathcal{F}}$ generated from $\mathcal{F}$, $\mathtt{Concs}(\mathcal{E})$ is consistent (resp. $\mathtt{Concs}(\mathcal{E})$ is closed) [7].

Definition 5 An extended logic program (ELP, for short) [8] is a set of rules of the form:

$$L_0 \leftarrow L_1, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n \quad (n \geq m \geq 0), \tag{2}$$

where each L_i is a literal, i.e. either an atom A or $\neg A$ preceded by classical negation $\neg$. *not* represents *negation as failure* (NAF). A literal preceded by *not* is called a NAF-literal. Let Lit_P be the set of all ground literals in the language of an ELP P. The semantics of an ELP is given by *answer sets* [8] (resp. *paraconsistent stable models* or *p-stable models*, for short [10]) defined as follows.

First, let P be a *not*-free ELP (i.e., for each rule $m = n$). Then, $S \subseteq Lit_P$ is an *answer set* of P if S is a minimal set satisfying the following two conditions **(i)**,**(ii)**: **(i)** For each ground instance of a rule $L_0 \leftarrow L_1, \ldots, L_m$ in P, if $\{L_1, \ldots, L_m\} \subseteq S$, then $L_0 \in S$. **(ii)** If S contains a pair of complementary literals L and $\neg L$, then $S = Lit_P$. Second, let P be any ELP and $S \subseteq Lit_P$. The *reduct* of P by S is a *not*-free ELP P^S which contains $L_0 \leftarrow L_1, \ldots, L_m$ iff there is a ground rule of the form (2) in P such that $\{L_{m+1}, \ldots, L_n\} \cap S = \emptyset$. Then S is an answer set of P if S is an answer set of P^S.

In contrast, p-stable models are regarded as answer sets defined without the condition **(ii)**. An answer set is *consistent* if it is not Lit_P; otherwise it is *inconsistent*. An ELP P is *consistent* if it has a consistent answer set; otherwise P is *inconsistent* under answer set semantics. On the other hand, a p-stable model is *inconsistent* if it contains a pair of complementary literals; otherwise it is *consistent*. For an ELP P, P is *consistent* if it has a consistent p-stable model; otherwise it is *inconsistent* under paraconsistent stable model semantics.

3. Consistency in Assumption-Based Argumentation

We discuss *ABA consistency* and *consistent* extensions in an ABA framework whose language contains explicit negation. The following theorem holds in ABA.

Theorem 1 [14] *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ \ } \rangle$ be a flat ABA framework and $\mathcal{E}$ be a complete argument extension of $AF_{\mathcal{F}}$ generated from $\mathcal{F}$.*

1. *$\mathcal{F}$ satisfies* the closure-property.
2. *$\mathcal{F}$ satisfies* the consistency-property *iff for every $\mathcal{E}$,* $\mathtt{Concs}(\mathcal{E})$ *is consistent iff for every $\mathcal{E}$,* $\mathtt{Concs}(\mathcal{E})$ *is not contradictory w.r.t. explicit negation $\neg$.*

Item 2 in Theorem 1 denotes that an ABA $\mathcal{F}$ satisfies the consistency-property iff $AF_{\mathcal{F}}$ satisfies the *direct consistency* postulate [2] under complete semantics. In contrast, ABA consistency is differently defined as follows.

Definition 6 ***(Consistent argument extensions)***[14] Given a flat ABA framework $\mathcal{F}$, let $\mathcal{E}$ be a complete argument extension of $AF_{\mathcal{F}}$ generated from $\mathcal{F}$. Then the extension $\mathcal{E}$ is said to be consistent if $\mathtt{Concs}(\mathcal{E})$ is not contradictory w.r.t. $\neg$; otherwise it is inconsistent.

Definition 7 ***(ABA consistency)*** [14] A flat ABA framework $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ is said to be *consistent* under σ semantics if $AF_{\mathcal{F}}$ generated from $\mathcal{F}$ has a consistent σ argument extension; otherwise it is inconsistent.

Note that if a flat ABA framework $\mathcal{F}$ satisfies the consistency-property, $\mathcal{F}$ is consistent under complete semantics, but not vice versa. ABA consistency can be also stated in terms of consistent assumption extensions based on the theorem shown below.

Proposition 1 ***(Consistent conflict-free sets of assumptions)*** *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ be a flat ABA framework and $A \subseteq \mathcal{A}$ be a conflict-free set of assumptions. Then A is* consistent *iff $CN_{\mathcal{R}}(A)$ is not contradictory w.r.t. $\neg$.*

Proof. *(i) Since $A \subseteq \mathcal{A}$ is conflict-free, it holds that $\nexists \alpha \in A$ such that $A' \vdash \bar{\alpha}$ for $A' \subseteq A$. (ii) Since $\mathcal{F}$ is flat, it holds that $Cn(A) \cap \mathcal{A} = A$ (iii) It holds that for a set $A \subseteq \mathcal{A}$, $CN_{\mathcal{R}}(A) = Cn(A)$. Then due to (i),(ii),(iii), there exists no assumption $\alpha \in \mathcal{A}$ such that $\{\alpha, \overline{\alpha}\} \subseteq Cn(A) = CN_{\mathcal{R}}(A)$, which means that $CN_{\mathcal{R}}(A)$ is not contradictory w.r.t. $\overline{}$. Hence a conflict-free set $A \subseteq \mathcal{A}$ is* consistent *iff $CN_{\mathcal{R}}(A)$ is not contradictory w.r.t. $\neg$.* □

The following is derived based on Proposition 1.

Corollary 1 ***(Consistent admissible set of assumptions/ consistent assumption extensions)*** *Let A be any one of an admissible set of assumptions and an assumption extension. Then $A \subseteq \mathcal{A}$ is* consistent *iff $CN_{\mathcal{R}}(A)$ is not contradictory w.r.t. $\neg$.*

Theorem 2 ***(ABA consistency)*** *A flat ABA framework $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ is* consistent *under σ semantics iff $\mathcal{F}$ has a consistent σ assumption extension; otherwise it is inconsistent.*

Proof. *($\Leftarrow$) Let $A \subseteq \mathcal{A}$ be a consistent σ assumption extension in $\mathcal{F}$. For A, there is a σ argument extension $\mathcal{E}$ in $\mathcal{F}$ such that $\mathcal{E} = \mathtt{Asms2Args}(A)$. Then it holds that $CN_{\mathcal{R}}(A) = Cn(A) = \{c \in \mathcal{L} \mid \exists K \vdash c, K \subseteq A\} = \{c \in \mathcal{L} \mid \exists K \vdash c \in \mathcal{E}\} = \mathtt{Concs}(\mathcal{E})$. Besides since the assumption extension $A \subseteq \mathcal{A} \subseteq \mathcal{L}$ is consistent, $CN_{\mathcal{R}}(A)$ is not contradictory w.r.t. $\neg$ due to Corollary 1. Therefore $\mathcal{F}$ has the consistent σ argument extension $\mathcal{E}$ since $\mathtt{Concs}(\mathcal{E}) = CN_{\mathcal{R}}(A)$ is not contradictory w.r.t. $\neg$. Hence $\mathcal{F}$ is consistent.*
($\Rightarrow$) The converse is also proved similarly. □

The following example illustrates the difference between satisfaction of the consistency-property and ABA consistency.

Example 1 The following ELP P_1 [2] expresses "*Married John*" [2] extended with the rule, $\neg wr \leftarrow not\ hw$:

$$P_1 = \{wr \leftarrow,\ \ go \leftarrow,\ \ m \leftarrow wr, not\ \neg m,\ \ b \leftarrow go, not\ \neg b,$$
$$hw \leftarrow m,\ \ \neg hw \leftarrow b,\ \ \neg b \leftarrow hw,\ \ \neg m \leftarrow \neg hw,\ \ \neg wr \leftarrow not\ hw\}.$$

P_1 has the unique consistent answer set $M_1 = \{wr, go, m, \neg b, hw\}$,
while it has two p-stable models, M_1 and $M_2 = \{wr, go, \neg m, b, \neg hw, \neg wr\}$,

[2] This ELP P_1 was inspired in a personal communication with Dr. Martin Caminada.

where M_1 is consistent but M_2 is inconsistent. Hence P_1 is consistent under answer set semantics as well as under paraconsistent stable model semantics.

In contrast, in the ABA framework $\mathcal{F}_{P_1} = \langle \mathcal{L}, P_1, \mathcal{A}, \overline{} \rangle$ instantiated with P_1 where $\mathcal{A} = \{not\ \neg m, not\ \neg b, not\ hw\}$, $\overline{not\ \neg m} = \neg m$, $\overline{not\ \neg b} = \neg b$, $\overline{not\ hw} = hw$, there are arguments and $attacks$ as follows:

$A_1 : \{\} \vdash wr$, $A_2 : \{\} \vdash go$, $A_3 : \{not\ \neg m\} \vdash m$, $A_4 : \{not\ \neg b\} \vdash b$,
$A_5 : \{not\ \neg m\} \vdash hw$, $A_6 : \{not\ \neg b\} \vdash \neg hw$, $A_7 : \{not\ \neg m\} \vdash \neg b$,
$A_8 : \{not\ \neg b\} \vdash \neg m$, $A_9 : \{not\ hw\} \vdash \neg wr$, $A_{10} : \{not\ \neg m\} \vdash not\ \neg m$,
$A_{11} : \{not\ \neg b\} \vdash not\ \neg b$, $A_{12} : \{not\ hw\} \vdash not\ hw$,
$attacks = \{(A_7, A_4), (A_7, A_6), (A_7, A_8), (A_7, A_{11}), (A_8, A_3), (A_8, A_5), (A_8, A_7),$
$(A_8, A_{10}), (A_5, A_9), (A_5, A_{12})\}$.

Then it has three complete argument extensions $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$ as follows:

$\mathcal{E}_1 = \{A_1, A_2, A_3, A_5, A_7, A_{10}\}$,
$\mathcal{E}_2 = \{A_1, A_2, A_4, A_6, A_8, A_9, A_{11}, A_{12}\}$,
$\mathcal{E}_3 = \{A_1, A_2\}$,

where $\texttt{Concs}(\mathcal{E}_1) = \{wr, go, m, hw, \neg\, b, not\ \neg m\}$,
$\texttt{Concs}(\mathcal{E}_2) = \{wr, go, b, \neg\, hw, \neg\, m, \neg wr, not\ \neg b, not\ hw\}$, $\texttt{Concs}(\mathcal{E}_3) = \{wr, go\}$.
Note that $\mathcal{E}_1, \mathcal{E}_2$ are stable extensions such that $\texttt{Concs}(\mathcal{E}_i) \cap Lit_{P_1} = M_i$ $(i = 1, 2)$.

Regarding classical negation $\neg$ contained in P_1 as explicit negation in $\mathcal{F}_{P_1}$, both $\mathcal{E}_1$ and $\mathcal{E}_3$ are consistent, while $\mathcal{E}_2$ is inconsistent. Therefore the ABA $\mathcal{F}_{P_1}$ is consistent under stable (resp. complete) semantics since it has the consistent stable extension $\mathcal{E}_1$, while $\mathcal{F}_{P_1}$ does not satisfy the consistency-property (i.e. violates the direct consistency postulate) since it has the inconsistent $\mathcal{E}_2$.

4. Consistency Required in ABA Applications

4.1. Counterexamples to Schulz and Toni's Theorems

In [11], an argument $K \vdash c$ in a flat ABA framework $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ is denoted by $(K, F) \vdash c$, where $F = \{\ell_N |\ \ell_N \leftarrow \in R\}$ is the set of heads of rules with an empty body singled out from the set $R \subseteq \mathcal{R}$ of inference rules used in the construction of the argument $K \vdash c$. Then in their approach, the relation $attacks$ is defined by using arguments whose form is $(K, F) \vdash c$ instead of $K \vdash c$ in (1) as follows:

- an argument $(K_1, F_1) \vdash c_1$ $attacks$ an argument $(K_2, F_2) \vdash c_2$ iff $c_1 = \overline{\alpha}$ for $\alpha \in K_2$.

Given an ELP P, they defined *the translated ABA framework* $ABA_P = \langle \mathcal{L}_P, \mathcal{R}_P$ $\mathcal{A}_P, \overline{} \rangle$, i.e. the ABA instantiated with P, where $NAF_P = \{not\ \ell | \ell \in Lit_P\}$, $\mathcal{L}_P = Lit_P \cup NAF_P$, $\mathcal{R}_P = P$, $\mathcal{A}_P = NAF_P$, and $\overline{not\ \ell} = \ell$ for every $not\ \ell \in \mathcal{A}_P$. For $S \subseteq Lit_P$, $\Delta_S = \{not\ \ell \in NAF_P | \ell \notin S\}$ is the set of NAF-literals. If S is an answer set of P, $S_{NAF} = S \cup \Delta_S$ is called an *answer set with NAF literals* of P. According to [11, Notation 2], $\vdash_{MP}$ denotes derivability using *modus ponens* on $\leftarrow$ as the only inference rule. When used on $P \cup \Delta_S$, $\vdash_{MP}$ treats NAF literals syntactically like facts. Then to justify why a literal belongs to an answer set of an ELP based on ABA, Schulz and Toni claimed their theorems and lemma [11, Theorems 1, 2 and Lemma 1] for a logic program P [3], i.e. a consistent ELP as follows.

[3] In [11, Section 2.1], it is described that "if not stated otherwise, we assume that logic programs are consistent".

(c1) **[11, Theorem 1].** Let P be a logic program and let $ABA_P = \langle \mathcal{L}_P, \mathcal{R}_P, \mathcal{A}_P, \overline{} \rangle$. Let X be a set of arguments in ABA_P and let $T = \{k \mid \exists (AP, FP) \vdash k \in X\}$ be the set of all conclusions of arguments in X. X is a stable extension of ABA_P if and only if T is an answer set with NAF literals of P.

(c2) **[11, Theorem 2].** Let P be a logic program and let $ABA_P = \langle \mathcal{L}_P, \mathcal{R}_P, \mathcal{A}_P, \overline{} \rangle$. Let $T \subseteq Lit_P$ be a set of classical literals and let $X = \{(AP, FP) \vdash k \mid AP \subseteq \Delta_T\}$ be the set of arguments in ABA_P whose assumptions are in Δ_T. T is an answer set of P if and only if X is a stable extension of ABA_P.

(c3) **[11, Lemma 1].** Let P be a consistent logic program and let $S \subseteq Lit_P$.

(i) S is an answer set of P if and only if $S = \{\ell \in Lit_P \mid P \cup \Delta_S \vdash_{MP} \ell\}$.

(ii) $S_{NAF} = S \cup \Delta_S$ is an answer set with NAF literals of P if and only if $S_{NAF} = \{k \mid P \cup \Delta_S \vdash_{MP} k\}$.

There exist counterexamples to their claims (c1), (c2) as follows.

Example 2 ***(Counterexamples to (c1), (c2))*** Consider the following ELP P_2 [14],

$$P_2 = \{\neg p \leftarrow not\ a, \quad a \leftarrow p, not\ b, \quad p \leftarrow, \quad b \leftarrow not\ a\},$$

where $Lit_{P_2} = \{a, b, p, \neg a, \neg b, \neg p\}$. P_2 has the unique consistent answer set $S_1 = \{a, p\}$, where $S_{1_{NAF}} = S_1 \cup \Delta_{S_1} = \{a, p, not\ b, not\ \neg a, not\ \neg b, not\ \neg p\}$ is the answer set with NAF literals of P_2, while P_2 has two p-stable models, $S_1 = \{a, p\}$ and $S_2 = \{\neg p, p, b\}$, where $\Delta_{S_2} = \{not\ a, not\ \neg a, not\ \neg b\}$. Thus S_1 is consistent but S_2 is inconsistent. Hence P_2 is consistent under answer set semantics as well as under paraconsistent stable model semantics.

In contrast, $ABA_{P_2} = \langle \mathcal{L}_{P_2}, P_2, \mathcal{A}_{P_2}, \overline{} \rangle$ is constructed from P_2, which has arguments and $attacks$ as follows:

$A_1 : (\{not\ a\}, \emptyset) \vdash \neg p,\ \ A_2 : (\{not\ b\}, \{p\}) \vdash a,\ \ A_3 : (\emptyset, \{p\}) \vdash p,$
$A_4 : (\{not\ a\}, \emptyset) \vdash b,\ \ A_5 : (\{not\ a\}, \emptyset) \vdash not\ a,\ \ A_6 : (\{not\ b\}, \emptyset) \vdash not\ b$
$A_7 : (\{not\ p\}, \emptyset) \vdash not\ p,\ \ \ A_8 : (\{not\ \neg a\}, \emptyset) \vdash not\ \neg a,$
$A_9 : (\{not\ \neg b\}, \emptyset) \vdash not\ \neg b,\ \ A_{10} : (\{not\ \neg p\}, \emptyset) \vdash not\ \neg p,$
$attacks = \{(A_1, A_{10}), (A_2, A_1), (A_2, A_4), (A_2, A_5), (A_3, A_7), (A_4, A_2), (A_4, A_6)\}.$

Then ABA_{P_2} has two stable extensions $\mathcal{E}_1$, $\mathcal{E}_2$ as follows.

$$\mathcal{E}_1 = \{A_2, A_3, A_6, A_8, A_9, A_{10}\}, \quad \mathcal{E}_2 = \{A_1, A_3, A_4, A_5, A_8, A_9\},$$

where $\mathtt{Concs}(\mathcal{E}_1) = \{a, p, not\ b, not\ \neg a, not\ \neg b, not\ \neg p\} = S_{1_{NAF}}$,
$\mathtt{Concs}(\mathcal{E}_2) = \{\neg p, p, b, not\ a, not\ \neg a, not\ \neg b\} = S_2 \cup \Delta_{S_2}$.

Hence $\mathcal{E}_1$ is consistent but $\mathcal{E}_2$ is not. Besides $\mathtt{Concs}(\mathcal{E}_i) = S_i \cup \Delta_{S_i}$ (or $\mathtt{Concs}(\mathcal{E}_i) \cap Lit_{P_2} = S_i$) holds for $\mathcal{E}_i$ and S_i ($i = 1, 2$). In the following, it is shown that claims (c1), (c2) do not hold for this consistent ELP P_2.

(1) Suppose that (c1) holds. Let X be the stable extension $\mathcal{E}_2$ of ABA_{P_2}. Then due to (c1), $T = \{k \mid \exists (AP, FP) \vdash k \in \mathcal{E}_2\} = \mathtt{Concs}(\mathcal{E}_2)$ should be the answer set with NAF literals of P_2. However $\mathtt{Concs}(\mathcal{E}_2) = S_2 \cup \Delta_{S_2}$ is not the answer set with NAF literals of P_2 since S_2 is not the answer set of P_2. Contradiction. Thus (c1) does not hold for P_2. □

(2) Suppose that (c2) holds. For $T = \{\neg p, p, b\}$ and $\Delta_T = \{not\ a, not\ \neg a, not\ \neg b\}$, $X = \{(AP, FP) \vdash k \mid AP \subseteq \Delta_T\} = \{A_1, A_3, A_4, A_5, A_8, A_9\} = \mathcal{E}_2$ is obtained, where $\mathcal{E}_2$ is the stable extension of ABA_{P_2}. Then due to (c2), $T = S_2$ should be the answer set of

P_2. However S_2 is not the answer set. Contradiction. Hence (c2) does not hold for P_2. $\square$

Remark: The ELP P_1 in Example 1 is also a counterexample to (c1), (c2) [11, Theorems 1,2]. (Details are omitted due to space limitations.)

The reason why their theorems [11, Theorems 1, 2] do not hold is that they proved them based on the claim (c3), to which there exists also a counterexample.

Example 3 ***(Counterexample to (c3) [11, Lemma 1])*** Consider the consistent ELP P_2 and $S_2 = \{\neg p, p, b\}$ in Example 2.

- Suppose that (c3) (i) holds. For P_2 and Δ_{S_2}, $\{\ell \in Lit_{P_2} \mid P_2 \cup \Delta_{S_2} \vdash_{MP} \ell\} = \{\neg p, p, b\} = S_2$ is derived. Then due to (c3) (i), S_2 should be the answer set of P_2. However S_2 is not the answer set of P_2 but the p-stable model. Contradiction. Thus (c3) (i) does not hold.
- Similarly we can easily show that (c3) (ii) does not hold. $\square$

4.2. Correspondence between Consistent Answer Sets and Consistent Stable Extensions

Two theorems [13, Theorems 3, 4] for an ELP were presented as *Extended Logic Programming as Argumentation*, whereas Schulz and Toni claimed (c1), (c2) [11, Theorems 1, 2] for a *consistent* ELP, i.e. the subclass of an ELP.

In [13, Theorems 3, 4], the following notations are used. Given an ELP P,

$$\mathcal{F}(P) = \langle \mathcal{L}_P, P, Lit_{not}, \overline{} \rangle$$

is the ABA framework instantiated with P, where $Lit_{not} = \{not\ L \mid L \in Lit_P\}$, $\mathcal{L}_P = Lit_P \cup Lit_{not}$ and $\overline{not\ L} = L$ for $not\ L \in Lit_{not}$. $AF_{\mathcal{F}}(P)$ denotes the abstract argumentation framework generated from the ABA $\mathcal{F}(P)$. For an ELP P, let

$$P_{tr} \stackrel{\text{def}}{=} P \cup \{L \leftarrow p, \neg p \mid p \in Lit_P,\ L \in Lit_P\},$$

be the ELP obtained from P by incorporating the *trivialization rules* [10]. Then $\mathcal{F}(P_{tr}) = \langle \mathcal{L}_P, P_{tr}, Lit_{not}, \overline{} \rangle$ is the ABA framework instantiated with P_{tr}, where $Lit_{P_{tr}} = Lit_P$ and $\mathcal{L}_{P_{tr}} = \mathcal{L}_P$. Besides for $M \subseteq Lit_P$, $\neg.CM = \{notL \mid L \in Lit_P \setminus M\}$ is the set of NAF literals.

Hence for an ELP P, $\mathcal{F}(P)$ (resp. Lit_{not}) corresponds to ABA_P (resp. $\mathcal{A}_P$) in [11], while for $M \subseteq Lit_P$, $M \cup \neg.CM$ (resp. $\neg.CM$) coincides with $M \cup \Delta_M$ (resp. Δ_M). Thus for an answer set M, $M \cup \neg.CM$ denotes $M_{NAF} = M \cup \Delta_M$ called *the answer set with NAF literals*. Theorems for an ELP are shown as follows.

Theorem 3 ***[13, Theorem 3].*** *Let P be an ELP. Then M is a p-stable model of P iff there is a stable extension $\mathcal{E}$ of $AF_{\mathcal{F}}(P)$ such that $M \cup \neg.CM = \mathtt{Concs}(\mathcal{E})$ (in other words, $M = \mathtt{Concs}(\mathcal{E}) \cap Lit_P$).*

Theorem 4 ***[13, Theorem 4].*** *Let P be an ELP. Then M is an answer set of P iff there is a stable extension $\mathcal{E}_{tr}$ of the ABA $\mathcal{F}(P_{tr})$ (or $AF_{\mathcal{F}}(P_{tr})$ such that $M \cup \neg.CM = \mathtt{Concs}(\mathcal{E}_{tr})$ (in other words, $M = \mathtt{Concs}(\mathcal{E}_{tr}) \cap Lit_P$).*

Example 2 illustrates that Theorem 3 holds for the p-stable model S_i of P_2 since $\mathtt{Concs}(\mathcal{E}_i) = S_i \cup \Delta_{S_i} = S_i \cup \neg.CS_i$ holds w.r.t. the stable extension $\mathcal{E}_i$ ($i = 1, 2$), while the following illustrates that Theorem 4 holds for answer sets of P_2.

Example 4 ***(Cont. Ex. 2)*** For P_2, P_{tr} is obtained as follows.
$P_{tr} = P_2 \cup \{L \leftarrow a, \neg a \mid L \in Lit_{P_2}\} \cup \{L \leftarrow b, \neg b \mid L \in Lit_{P_2}\} \cup \{L \leftarrow p, \neg p \mid L \in Lit_{P_2}\}$.
Then the ABA $\mathcal{F}(P_{tr})$ (i.e. $ABA_{P_{tr}}$) has the unique stable extension $\mathcal{E}_{tr} = \mathcal{E}_1$ such that $\mathtt{Concs}(\mathcal{E}_{tr}) = S_1 \cup \neg.CS_1 = S_1 \cup \Delta_{S_1} = S_{1_{NAF}}$ for the answer set with NAF literals $S_{1_{NAF}}$, where S_1 is the unique consistent answer set of P_2.

In what follows, we prove and present the correct theorems against their claims. First of all, we provide the following lemmas regarding a consistent ELP.

Lemma 1 *Let P be an ELP. M is a* consistent *answer set of P iff there is a* consistent *p-stable model M of P.*
Proof. *($\Leftarrow$) Let M be a consistent p-stable model of P. Then M does not contain a pair of complementary literals. Since M is also a p-stable model of the* reduct P^M *according to Def. 5, M is a minimal set satisfying the condition (i) for P^M which is the not-free ELP. Then since M does not contain a pair of complementary literals, M is also a minimal set satisfying both conditions (i) and (ii) for P^M. This denotes that M is the answer set of P^M which does not contain a pair of complementary literals. Thus M is the answer set of P^M and it is not Lit_P. Hence since the answer set M of P^M which is not Lit_P is the answer set of P, M is the consistent answer set of P.*
($\Rightarrow$) The converse is proved in a similar way. □

The following corollary is the direct result of Lemma 1.

Corollary 2 *An ELP P is* consistent *under answer set semantics iff P is* consistent *under paraconsistent stable model semantics.*

Lemma 2 *Let P be a* consistent *ELP. If M is an answer set of P, M is a p-stable model of P, but not vice versa.*
Proof. *($\Rightarrow$) Since P is consistent, its answer set M is consistent. Thus due to Lemma 1, M is a p-stable model of P.*
($\Leftarrow$) A consistent ELP P has a consistent p-stable model which is the answer set of P. Moreover it may have an inconsistent p-stable model M containing a pair of complementary literals L and $\neg L$. Then suppose that such inconsistent p-stable model M is also the answer set of P. Since M is the answer set of P, M is a minimal set satisfying the condition (i),(ii) in Def. 5 for the reduct P^M. Thus M is Lit_P due to (ii) since M contains a pair of complementary literals. However P has a consistent answer set $S \subset Lit_P$ because P is consistent. Thus M which is Lit_P is not minimal. Hence M is not the answer set of P. Contradiction. □

Hereby given a consistent ELP, we can obtain the following theorems.

Theorem 5 *Let P be a* consistent *ELP. Then M is an answer set of P iff there is a* consistent *stable extension $\mathcal{E}$ of the ABA framework $\mathcal{F}(P)$ (or $AF_{\mathcal{F}}(P)$) such that $M \cup \neg.CM = \mathtt{Concs}(\mathcal{E})$.*
Proof. *($\Leftarrow$) Let $\mathcal{E}$ be a consistent stable extension of the ABA $\mathcal{F}(P)$. Then $\mathtt{Concs}(\mathcal{E})$ is consistent, i.e. not contradictory w.r.t. $\neg$. Due to Theorem 3, for the stable extension $\mathcal{E}$, there is the p-stable model M of P such that $M \cup \neg.CM = \mathtt{Concs}(\mathcal{E})$. Since $\mathtt{Concs}(\mathcal{E})$ does not contain a pair of complementary literals, $M \cup \neg.CM$ as well as the p-stable model M are consistent. Hence due to Lemma 1, M is the consistent answer set of P.*
($\Rightarrow$) The converse is also proved in a similar way. □

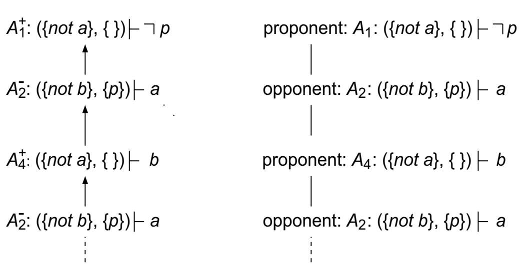

Figure 1. The admissible dispute tree $\mathcal{T}_{\mathcal{E}_2}(A_1)$ (right) vs. the positive attack tree $attTree^{+}_{\mathcal{E}_2}(A_1)$ (left) in Ex. 5

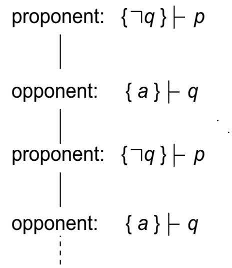

Figure 2. The admissible dispute tree $\mathcal{T}$ for $\{\neg q\} \vdash p$ in Ex. 6

Theorem 6 *Let P be a* consistent *ELP. If M is an answer set of P, there is a stable extension $\mathcal{E}$ of the ABA framework $\mathcal{F}(P)$ such that $M \cup \neg.CM = \mathtt{Concs}(\mathcal{E})$, but not vice versa.*

Proof. *This is proved based on Lemma 2 and Theorem 3.* □

Corollary 3 *Let P be a* consistent *ELP. $\mathcal{E}$ is a* consistent *stable extension of the ABA framework $\mathcal{F}(P)$ iff $\mathcal{E}$ is a stable extension of the ABA framework $\mathcal{F}(P_{pr})$.*

Theorem 5 and Theorem 6 state that there is a one-to-one correspondence between answer sets of a *consistent* ELP P and (*not* stable extensions but) *consistent* stable extensions of the ABA $\mathcal{F}(P)$ contrary to their claims (c1), (c2).

As for rationality postulates, the following theorem generally holds as illustrated in Example 1, which implies that Schulz and Toni's theorems are incorrect.

Theorem 7 *Let P be a* consistent *ELP. Then the ABA framework $\mathcal{F}(P)$ instantiated with P is consistent under complete (resp. stable) semantics, while it is not guaranteed to satisfy the consistent property or the direct consistency postulate.*

Proof. *There is an answer set of P. Then there is a consistent stable extension of $\mathcal{F}(P)$ based on Theorem 5. Hence $\mathcal{F}(P)$ is consistent under those semantics. Similarly there may be an inconsistent p-stable model of P. Then $\mathcal{F}(P)$ may have an inconsistent stable extension based on Theorem 3. Thus the latter is proved.* □

4.3. Consistency for Admissible Dispute Trees

Admissibility is defined for abstract (resp. concrete) dispute trees [5]. However *consistency* has not been taken into account for admissible dispute trees so far even though the following serious consistency problem may arise in ABA containing explicit negation.

Example 5 ***(Cont. Ex. 2)*** Consider ABA_{P_2} where classical negation $\neg$ in P_2 is regarded as explicit negation. In Figure 1, the left is the *positive Attack tree* $attTree^{+}_{\mathcal{E}_2}(A_1)$ of the argument $A_1 : (\{not\ a\}, \emptyset) \vdash \neg p$ with respect to the stable extension $\mathcal{E}_2 = \{A_1, A_3, A_4, A_5, A_8, A_9\}$ of ABA_{P_2}, while the right is the admissible abstract dispute tree $\mathcal{T}_{\mathcal{E}_2}(A_1)$ for A_1 translated from $attTree^{+}_{\mathcal{E}_2}(A_1)$ according to [11, Lemma 11]. Though there exists a fact p, i.e. $p \leftarrow\ \in P_2$ in ABA_{P_2}, the belief $\neg p$ is concluded to be admissible since the root of $\mathcal{T}_{\mathcal{E}_2}(A_1)$ is labelled with A_1 whose claim is $\neg p$, that implies contradiction. In fact, its defence set $\mathcal{D}(\mathcal{T}_{\mathcal{E}_2}(A_1)) = \{not\ a\}$ is inconsistent since $CN_{P_2}(\{not\ a\}) = \{\neg p, p, b, not\ a\}$ is contradictory w.r.t. $\neg$.

To detect and avoid such anomaly in ABA whose language contains explicit negation, we introduce the notion of *consistency* into admissible dispute trees.

Definition 8 ***(Consistent admissible dispute trees)*** Given a flat ABA framework $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$, an admissible abstract (resp. concrete) dispute tree $\mathcal{T}$ is *consistent* if its defence set $\mathcal{D}(\mathcal{T})$ is *consistent*; otherwise it is *inconsistent*.

Proposition 2 ***(Consistent defence sets)*** *The defence set $\mathcal{D}(\mathcal{T}) \subseteq \mathcal{A}$ of an admissible dispute tree $\mathcal{T}$ is consistent iff $CN_{\mathcal{R}}(\mathcal{D}(\mathcal{T}))$ is not contradictory w.r.t. $\neg$.*

Proof. *This is proved due to Corollary 1 since $\mathcal{D}(\mathcal{T})$ is admissible.* □

A simplified assumption-based framework [5] is often used to illustrate an admissible dispute tree without stating consistency. The following ensures its consistency.

Proposition 3 *A simplified assumption-based framework (a simplified ABA, for short) [5] is an ABA framework $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$, where $\mathcal{F}$ is flat, all sentences in $\mathcal{L}$ are atoms $p, q, \ldots$ or negations of atoms $\neg p, \neg q, \ldots$ and $\overline{p} = \neg p$ for $p \in \mathcal{A}$ (resp. $\overline{\neg p} = p$ for $\neg p \in \mathcal{A}$). Then any admissible abstract (resp. concrete) dispute tree $\mathcal{T}$ in $\mathcal{F}$ is consistent.*

Proof. *Let $\alpha = p$ for $p \in \mathcal{A}$ (resp. $\alpha = \neg p$ for $\neg p \in \mathcal{A}$). Then $\{\alpha, \overline{\alpha}\} = \{p, \neg p\}$ for $p \in \mathcal{A}$ (resp. $\neg p \in \mathcal{A}$) is derived, which means that contradictoriness w.r.t. $\neg$ in $\mathcal{F}$ becomes contradictoriness w.r.t. $\overline{}$ in $\mathcal{F}$. Now let $\mathcal{T}$ be an admissible dispute tree in $\mathcal{F}$. Since $\mathcal{D}(\mathcal{T})$ is admissible, it is conflict-free. Besides $\mathcal{F}$ is flat. Then due to the proof of Proposition 1, $CN_{\mathcal{R}}(\mathcal{D}(\mathcal{T}))$ is not contradictory w.r.t. $\overline{}$ in $\mathcal{F}$. Hence $CN_{\mathcal{R}}(\mathcal{D}(\mathcal{T}))$ is also not contradictory w.r.t. $\neg$ in $\mathcal{F}$. Therefore any admissible dispute tree $\mathcal{T}$ in $\mathcal{F}$ is consistent since any $\mathcal{D}(\mathcal{T})$ is consistent based on Proposition 2.* □

Proposition 3 denotes that the consistency problem shown in Example 5 never arises in a simplified ABA. However the ABA $\mathcal{F}(P)$ (i.e. ABA_P) instantiated with an ELP P is not a simplified ABA.

Example 6 Consider the ABA $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$, where $\mathcal{R} = \{p \leftarrow \neg q,\ q \leftarrow a, \neg p \leftarrow\}$, $\mathcal{A} = \{\neg q, a\}$, $\overline{\neg q} = q$ and $\overline{a} = p$. $\mathcal{F}$ is not a simplified ABA. It has three complete extensions $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_3$ such that $\texttt{Concs}(\mathcal{E}_1) = \{q, \neg p, a\}$, $\texttt{Concs}(\mathcal{E}_2) = \{p, \neg p, \neg q\}$, $\texttt{Concs}(\mathcal{E}_3) = \{\neg p\}$, where $\mathcal{E}_1$, $\mathcal{E}_2$ are stable extensions. $\mathcal{E}_1$, $\mathcal{E}_3$ are consistent but $\mathcal{E}_2$ is not. Then $\mathcal{F}$ is consistent under stable (resp. complete) semantics, while it does not satisfy the consistency property. Figure 2 shows the admissible abstract dispute tree $\mathcal{T}$ for the argument $\{\neg q\} \vdash p$. Its defence set $\mathcal{D}(\mathcal{T}) = \{\neg q\}$ is inconsistent since $CN_{\mathcal{R}}(\{\neg q\}) = \{p, \neg p, \neg q\}$ is contradictory w.r.t. $\neg$. Hence $\mathcal{T}$ is inconsistent though it is admissible. In contrast, the admissible abstract dispute tree $\mathcal{T}'$ for $\{a\} \vdash q$ is consistent since $\mathcal{D}(\mathcal{T}') = \{a\}$ is consistent due to $CN_{\mathcal{R}}(\{a\}) = \{q, \neg p, a\}$.

4.4. The Necessary and Sufficient Condition to Guarantee ABA Consistency

We show the condition to guarantee ABA consistency. Given an ABA framework $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ whose set of arguments is finite, let Π be the ELP translated from $\mathcal{F}$ with no hypotheses (i.e. $\mathcal{H} = \emptyset$) defined in [14, Definition 13].

Then based on [14, Lemma 1], the *requirement* that the ELP Π (resp. $\Pi \cup \{\leftarrow undec(X)\}$ where $\mathcal{H} = \emptyset$ [14] should be consistent under answer set semantics is the *necessary* and *sufficient* condition that guarantees ABA consistency such that the ABA framework $\mathcal{F}$ is consistent under complete (resp. stable) semantics.

5. Related Work and Conclusion

Dung and Thang presented the *sufficient* condition referred to as the *ab-self-contradiction axiom* that guarantees closure- and consistency-properties in a flat ABA framework [7].

On the other hand, in [12], though it is shown that not the standard ABA but the generalized ABA mapped from a defeasible framework under some assumptions satisfies the closure and consistency postulates, Toni presented no results about satisfaction of those postulates in a standard flat ABA framework.

In this paper, we showed counterexamples to Schulz and Toni's theorems [11, Theorems 1, 2]. Then against their claims, we presented Theorems 5 and 6 showing that answer sets of a consistent ELP are captured by *not* stable extensions but *consistent* stable extensions of the ABA instantiated with the ELP. Theorem 7 shows such ABA instantiated with a consistent ELP is not ensured to satisfy the consistency postulate, that implies incorrectness of their theorems. We proposed the novel notion of consistency for admissible dispute trees to avoid anomalies in ABAs containing explicit negation. Finally we showed the condition to ensure ABA consistency. Our future work is to implement the method to compute consistent reasoning over ABA in answer set programming [8] (e.g. by using the ELP Π with $\mathcal{H} = \emptyset$ based on [14, Lemma 1]).

Acknowledgments This work was supported by KAKENHI (Grant-in-Aid for Scientific Research (S) 17H06103).

References

[1] A. Bondarenko, P.M. Dung, R.A. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93:63–101, 1997.

[2] M. Caminada and L. Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171 (5-6):286–310, 2007.

[3] M. Caminada, S. Sá, J.F.L. Alcântara, and W. Dvořá. On the difference between assumption-based argumentation and abstract argumentation. In *Proc. BNAIC'2013*, pages 25–32, 2013.

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

[5] P.M. Dung, R.A. Kowalski, and F. Toni. Dialectic proof procedures for assumption-based, admissible argumentation. *Artificial Intelligence*, 170(2):114-159, 2006.

[6] P.M. Dung, R.A. Kowalski, and F. Toni. Assumption-based argumentation. In *Argumentation in Artificial Intelligence*, pages 199–218, Springer, 2009.

[7] P.M. Dung and P.M. Thang. Closure and consistency in logic-associated argumentation. *Journal of Artificial Intelligence Research*, 49:79-109, 2014.

[8] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3/4):365–385, 1991.

[9] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument & Computation*, 1(2):93–124, 2010.

[10] C. Sakama and K. Inoue. Paraconsistent stable semantics for extended disjunctive programs. *Journal of Logic and Computation*, 5(3):265–285, 1995.

[11] C. Schulz and F. Toni. Justifying answer sets using argumentation. *Theory and Practice of Logic Programming*, 16(1):59–110, 2016.

[12] F. Toni. Assumption-Based Argumentation for Closed and Consistent Defeasible Reasoning. In *Proc. JSAI'2007*, volume 4914 of LNCS, pages 390–402, Springer, 2007.

[13] T. Wakaki. Assumption-based argumentation equipped with preferences and its application to decision-making, practical reasoning, and epistemic reasoning. *Computational Intelligence*, 33(4):706–736, 2017.

[14] T. Wakaki. Extended Abduction in Assumption-Based Argumentation. In *Proc. IEA/AIE'2019*, volume 11606 of LNAI, pages 593–607, Springer, 2019.