Argument & Computation -1 (2023) 1–45
DOI 10.3233/AAC-220019
IOS Press

Assumption-based argumentation for extended disjunctive logic programming and its relation to nonmonotonic reasoning

Toshiko Wakaki
Shibaura Institute of Technology, 307 Fukasaku, Minuma-ku, Saitama-city, Saitama, 337–8570, Japan
E-mail: twakaki@shibaura-it.ac.jp

Abstract. The motivation of this study is that Reiter's default theory as well as assumption-based argumentation frameworks corresponding to default theories have difficulties in handling disjunctive information, while a disjunctive default theory (*ddt*) avoids them. This paper presents the semantic correspondence between generalized assumption-based argumentation (ABA) and extended disjunctive logic programming as well as the correspondence between ABA and nonmonotonic reasoning approaches such as disjunctive default logic and prioritized circumscription. To overcome the above-mentioned difficulties of ABA frameworks corresponding to default theories, we propose an assumption-based framework (ABF) translated from an extended disjunctive logic program (EDLP) since an EDLP can be translated into a *ddt*. Our ABF incorporates explicit negation and the connective of disjunction "|" to Heyninck and Arieli's ABF induced by a disjunctive logic program. In this paper, first, we show how arguments are constructed from disjunctive rules in our proposed ABF. Then, we show the correspondence between answer sets of an EDLP P and stable extensions of the ABF translated from P with trivialization rules. After defining rationality postulates, we show answer sets of a consistent EDLP are captured by consistent stable extensions of the translated ABF with no trivialization rules. Finally, we show the correspondence between ABA and disjunctive default logic (resp. prioritized circumscription). The relation between ABA and possible model semantics of EDLPs is also discussed.

Keywords: Assumption-based argumentation, extended disjunctive logic programs, disjunctive default logic, arguments, prioritized circumscription

1. Introduction

1.1. Background

Disjunctive information is often required in reasoning and argumentation to solve problems in our daily life. In nonmonotonic reasoning, Gelfond et al. [16] proposed *disjunctive default logic* as a generalization of Reiter's default logic [27] to overcome problems of default logic in handling disjunctive information. To this end, they use the symbol "|" as the connective of disjunction in a *disjunctive default theory* instead of the classical "$\vee$" used in a default theory, where $P \vee Q$ means "$P \vee Q$ is known", while $P \mid Q$ means "P is known or Q is known".[1] They also showed that an extended disjunctive logic program (EDLP) [15] which may include disjunction using the connective "|" in rule head as well as two kinds of negation (i.e. classical negation "$\neg$" and negation-as-failure "*not*") can be translated into a disjunctive default theory. In contrast, in the context of formal argumentation, Beirlaen et al. [2,3]

[1] Semantically, the latter requires an extension to contain one of the two disjunctive terms, rather than the disjunction [16].

presented the extended ASPIC$^+$ framework where disjunctive reasoning is integrated in structured argumentation with defeasible rules [25] by incorporating reasoning by cases inference scheme [2]. In their framework, an argument is allowed to have a disjunctive conclusion, while disjunction is expressed by using the classical connective "$\vee$". However, they did not show the relationship between their framework and other approaches in nonmonotonic reasoning such as a disjunctive default theory. Consider the following example [3] shown by them.

Example 1 (**Kyoto protocol**). *There are two candidates for an upcoming presidential election. The candidates had a debate in the capital. They were asked what measures are to be taken in order for the country to reach the Kyoto protocol objectives for reducing greenhouse gas emissions. The first candidate, a member of the purple party, argued that if she wins the election, she will reach the objectives by supporting investments in renewable energy. The second candidate, a member of the yellow party, argued that if she wins the election, she will reach the objectives by supporting sustainable farming methods. We have reasons to believe that, no matter which candidate wins the election, the Kyoto protocol objectives will be reached. If the purple candidate wins, she will support investments in renewable energy ($p \Rightarrow r$), which would in turn result in meeting the Kyoto objectives ($r \Rightarrow k$). Similarly, if the yellow candidate wins, she will support sustainable farming methods ($y \Rightarrow f$), which would result in meeting the Kyoto objectives ($f \Rightarrow k$). Since one of the two candidates is going to win ($p \vee y$), we can reason by cases to conclude that the Kyoto objectives will be reached (k).*

Beirlaen et al. [3] represented the information shown above in terms of the knowledge base $\mathcal{K}_1 = (\{p \Rightarrow r,\ r \Rightarrow k,\ y \Rightarrow f,\ f \Rightarrow k\}, \{p \vee y\})$ which consists of *defeasible* rules in ASPIC$^+$ [24,25] and the classical formula $p \vee y$ as facts. Their formulation derives the intended result k, however, a problem happens when $\mathcal{K}_1$ is translated into a default theory. In fact, $\psi \Rightarrow \phi$ encodes the normal default $\frac{\psi:\phi}{\phi}$, then $\mathcal{K}_1$ is expressed by the default theory D_1:

$$D_1 = \left\{\frac{p:r}{r}, \frac{r:k}{k}, \frac{y:f}{f}, \frac{f:k}{k},\ p \vee y\right\}.$$

D_1 has a single extension, consisting of the disjunction $p \vee y$ and its logical consequences, where the four defaults "don't work". Thus the result is contrary to what we would expect.

Moreover, regarding ABA as another structured argumentation system, Lehtonen et al. [19] recently presented the ABA framework instantiated with a propositional default theory. However the ABA framework corresponding to D_1 which is constructed according to Lehtonen et al.'s definition cannot derive the expected result k under the stable (resp. grounded) semantics as well (see details in Section 6).

Notice that the disjunctive information s.t. "*one of the two candidates is going to win*" is expressed by $p \vee y$ in $\mathcal{K}_1$ and D_1. In a precise sense, however, it means that the election will yield a result s.t. either "that the purple candidate wins is known" or "that the yellow candidate wins is known". Hence it should be expressed by $p \mid y$ rather than $p \vee y$, that requires an extension to contain *one of the two disjunctive terms* rather than the disjunction due to [16]. Therefore, to solve the aforementioned problem arising in D_1, let us represent the information by the disjunctive default theory D_2:

$$D_2 = \left\{\frac{p:r}{r}, \frac{r:k}{k}, \frac{y:f}{f}, \frac{f:k}{k},\ p \mid y\right\},$$

or the associated EDLP $P_1 = \{r \leftarrow p, not\ \neg r,\quad k \leftarrow r, not\ \neg k,\quad f \leftarrow y, not\ \neg f,\quad k \leftarrow f, not\ \neg k,\quad p \mid y \leftarrow\}$ [16]. Then P_1 has two answer sets $S_1 = \{p, r, k\}$ and $S_2 = \{y, f, k\}$, while

D_2 has two extensions E_1 and E_2 such that $k \in S_i \subseteq E_i$ $(i = 1, 2)$, where E_i consists of atoms from S_i as well as the logical consequences derived from them. These results agree with our expectation that the Kyoto protocol objectives will be reached no matter which candidate wins the election. $\square$

In regard to the relation between assumption-based argumentation and (disjunctive) default logic, Bondarenko et al. firstly showed in [5, Theorem 3.16] that there is a one-to-one correspondence between extensions of Reiter's default theory and stable extensions of the corresponding assumption-based framework (ABF,[2] for short). This denotes that the expected result k of the Kyoto protocol problem shown above is never obtained from stable extensions of their ABF corresponding to D_1 based on their theorem. Besides, Bondarenko et al. [5] showed nothing about the relationship between ABA and disjunctive default logic. Hence, to solve such problems of ABFs corresponding to default theories in handling disjunctive information, it is required to find the relationship between ABFs and disjunctive default theories.

Recently, Heyninck and Arieli [18] proposed a generalized assumption-based framework induced by a disjunctive logic program (DLP), where disjunction using the connective "$\vee$" in rule head as well as one kind of negation, i.e. negation-as-failure (NAF) are allowed to appear. Hence though their ABF induced by a DLP has a *contrariness* operator $\overline{}$ such that $\overline{not\ p} = p$ for every atom p, its language does not contain explicit negation. Then they showed a one-to-one correspondence between the *stable models* of a DLP [26] and the stable *assumption* extensions of the ABF induced by a DLP. In their work, however, there are several open problems left to be explored. First, the semantic relationship between their ABF induced by a DLP and approaches of nonmonotonic reasoning such as a disjunctive default theory is not considered in [18]. Second, they did not show how to construct an argument from disjunctive rules in a DLP nor took account of *argument* extensions in their ABF.

In logic programming, EDLPs [15] were proposed by Gelfond and Lifschitz to extend DLPs for knowledge representation by not only allowing classical negation (i.e. explicit negation) along with negation-as-failure but also using "|" instead of "$\vee$". At the same time, it is shown in [16] that an EDLP can be embedded into a disjunctive default theory (*ddt*, for short) which uses "|" as the connective of disjunction. In contrast, in formal argumentation, generally an assumption-based framework is capable of containing explicit negation in its language [9,10,12], but the ABFs corresponding to Reiter's default theories have difficulties in handling disjunctive information. To our best knowledge, however, there is no study to show the relationship between ABFs and EDLPs as well as the relationship between ABFs and *ddt*s to overcome such difficulties of ABFs discussed above.

1.2. Purpose of this paper

The purpose of this paper is first to investigate the semantic relationship between ABFs and EDLPs as well as the relationships between ABFs and other approaches in nonmonotonic reasoning (e.g. disjunctive default logic [16], prioritized circumscription [20,23]) that have not been studied, and second to show how to construct an argument in ABFs from disjunctive rules in (E)DLPs. We use the answer set semantics [15] and the paraconsistent stable model semantics [28] of EDLPs *neither* of which is *two-valued* for characterizing stable extensions of ABFs translated from EDLPs.

Regarding ABFs whose languages contain explicit negation, however, we should pay attention to avoid consistency problems which sometimes have occurred in applications of argumentation due to explicit

[2] In this paper, we use the abbreviation "ABF" to primarily denote *assumption-based framework* but sometimes *ABA framework* in different approaches [5,10,18,19,32,33] though it may refer to the respectively defined framework based on ABA in each approach.

negation contained in the language. To this end, so far, *rationality postulates* [6] were proposed as principles which rule-based argumentation systems should satisfy to avoid anomalous outcomes. In fact, in structured argumentation systems such as ASPIC$^+$ and ABA frameworks whose languages contain explicit negation, conditions under which each system satisfies rationality postulates were proposed [1,12,24,25].

As for recent ABA applications containing explicit negation, Schulz and Toni [31] proposed the approach of justifying answer sets of an extended logic program (ELP) using argumentation. In their approach, they used the ABA framework ABA_P instantiated with an ELP P. Though an ELP may contain classical negation, they took account of neither rationality postulates [6,12] nor consistency in ABA, and they claimed in their theorems [31, Theorem 1 and Theorem 2] that there is a one-to-one correspondence between answer sets of a consistent ELP P and the stable extensions of ABA_P. In other words, their theorems claim that any ABA framework instantiated with a consistent ELP will never produce inconsistent stable extensions. However, there exist counterexamples to their theorems as shown in [33]. This means that anomalous outcomes may be obtained in applications of argumentation based on their theorems (e.g., there exists the inconsistent stable extension in ABA_P instantiated with the given consistent ELP P which expresses the knowledge of the slightly modified *Married John Example* [33, Example 1], [6]).

Therefore, to achieve the above-mentioned purpose of this study, this paper proposes an assumption-based framework translated from an EDLP, which incorporates explicit negation as well as the connective "|" instead of "$\vee$" in Heyninck and Arieli's ABF induced by a DLP while avoiding consistency problems that arise in Schulz and Toni's approach. Contributions of this study are as follows:

First, we define an argument in the ABF translated from a given EDLP which is constructed from disjunctive rules of the EDLP based on three inference rules provided in our ABF. Second, we show not only a one-to-one correspondence between *p-stable models* of an EDLP P and stable *argument* extensions (resp. stable *assumption* extensions) of the ABF translated from P but also a one-to-one correspondence between *answer sets* of an EDLP P and stable *argument* extensions (resp. stable *assumption* extensions) of the ABF translated from P with trivialization rules. Third, since our ABF incorporates explicit negation $\neg$, we define rationality postulates and consistency in our ABFs to avoid anomalous outcomes. Then we show answer sets of a consistent EDLP can be captured by consistent stable extensions of the translated ABF with no trivialization rules. This is useful for ABA applications containing explicit negation. Fourth, we show the new results about the relationship between ABA and disjunctive default logic [16] which enables us to overcome the difficulties of the ABF corresponding to a default logic in handling disjunctive information as well as the relationship between ABA and prioritized circumscription [20,23]. These results have not been shown so far to the best of our knowledge. Finally, we show how the possible model semantics of EDLPs [30] which is different from answer set semantics [15] is also captured by our ABFs translated from EDLPs. Appendix shows the correspondence between the semantics of ELPs [15,28] and stable *assumption* extensions of the associated ABA frameworks.

This paper is an extended and revised version of the paper [34], where not only Theorems 13, 14 and 15 in Section 4.1 but also Section 4.2, Section 5 and Appendix are newly introduced. Every proof sketch in [34] is replaced with a full proof while introducing new propositions, lemmas and two tables in this paper. Revisions are made throughout the paper and new considerations[3] are often added. The rest of this paper is organized as follows. Section 2 shows preliminaries where new propositions proved in Appendix are mentioned. Section 3 presents an ABF translated from an EDLP, the definition of its argument,

[3]For example, the difference between DLPs and NDPs as shown in Eample 4, the interesting results under the skeptical semantics (e.g. grounded) in Example 8, and the relation between our ABF and a standard ABA as shown in Proposition 8, etc.

the semantic relationship between an EDLP and the corresponding ABF, the notion of consistency in ABFs, and the semantic relationship between a consistent EDLP and the corresponding consistent ABF. Section 4 shows the relationships between our proposed ABFs and nonmonotonic reasoning formalisms such as disjunctive default logic and prioritized circumscription. Section 5 shows the correspondence between the possible model semantics of EDLPs and ABA semantics of our ABFs. Section 6 discusses related work. Section 7 presents conclusions. Appendix contains new propositions relating ELPs and ABFs.

2. Preliminaries

2.1. Assumption-based argumentation

An ABA framework [5,9,10] is a tuple $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$, where $(\mathcal{L}, \mathcal{R})$ is a deductive system, consisting of a formal language (a set of sentences) $\mathcal{L}$ and a set $\mathcal{R}$ of inference rules of the form: $b_0 \leftarrow b_1, \dots, b_m$ ($b_i \in \mathcal{L}$ for $0 \leqslant i \leqslant m$), $\mathcal{A} \subseteq \mathcal{L}$ is a (non-empty) set of *assumptions*, and $\overline{}$ is a total mapping from $\mathcal{A}$ into $\mathcal{L}$, which we call a contrariness operator. $\overline{\alpha}$ is referred to as the *contrary* of $\alpha \in \mathcal{A}$. For a rule $r \in \mathcal{R}$ of the form $b_0 \leftarrow b_1, \dots, b_m$, let the head be $head(r) = b_0$ and the body be $body(r) = \{b_1, \dots, b_m\}$. We enforce that ABA frameworks are *flat*, namely assumptions in $\mathcal{A}$ do not appear in the heads of rules in $\mathcal{R}$. In an ABA framework, *arguments* and *attacks* are defined as follows.

Definition 1 ([10,11]). Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ be an ABA framework. An *argument for the conclusion (or claim)* $c \in \mathcal{L}$ *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 the special symbol $\tau \notin \mathcal{L}$ representing "true", such that

- the root is labelled by c,
- for every node N
  - if N is a leaf then N is labelled by an assumption in $\mathcal{A}$ or by τ;
  - if N is not a leaf and b_0 is the label of N, then there is an inference rule $b_0 \leftarrow b_1, \dots, b_m (m \geqslant 0)$ and either $m = 0$ and the child of N is labelled by τ or $m > 0$ and N has m children, labelled by $b_1, \dots, b_m$ respectively,
- K is the set of all assumptions labelling the leaves.

attacks between arguments and *attacks* between sets of assumptions are defined respectively as follows.

- *An argument* $K_1 \vdash c_1$ *attacks an argument* $K_2 \vdash c_2$ iff $c_1 = \overline{\alpha}$ for some $\alpha \in K_2$.
- For $\Delta, \Delta' \subseteq \mathcal{A}$, and $\alpha \in \mathcal{A}$, Δ *attacks* α iff Δ enables the construction of an argument for conclusion $\overline{\alpha}$. Accordingly, Δ *attacks* Δ' if Δ *attacks* some $\alpha \in \Delta'$.

A set of arguments *Args* is *conflict-free* iff $\nexists A, B \in Args$ such that A *attacks* B. *Args defends* an argument A iff each argument that attacks A is attacked by an argument in *Args*. On the other hand, a set of assumptions Δ is *conflict-free* iff Δ does not attack itself. Δ *defends* $\alpha \in \mathcal{A}$ iff each $\Delta' \subseteq \mathcal{A}$ that attacks α is attacked by Δ.

Let $AF_{\mathcal{F}} = (AR, attacks)$ be the abstract argumentation (AA) framework [8] generated from an ABA framework $\mathcal{F}$, where AR is the set of all arguments such that $a \in AR$ iff an argument $a : K \vdash c$ is in $\mathcal{F}$, and $(a, b) \in attacks$ in $AF_{\mathcal{F}}$ iff a *attacks* b in $\mathcal{F}$ [11].

Let $\sigma \in \{complete, preferred, grounded, stable, ideal\}$ be the name of the argumentation semantics. The ABA semantics is given by σ *argument extensions* as well as by σ *assumption extensions*. A σ *argument extension* $Args \subseteq AR$ and a σ *assumption extension* $\Delta \subseteq \mathcal{A}$ are defined respectively as follows.

$Args \subseteq AR$ 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 attacks every argument in $AR \setminus Args$; an *ideal* argument extension iff it is a (subset-)maximal complete argument extension that is contained in each preferred argument extension. In contrast, $\Delta \subseteq \mathcal{A}$ is: *admissible* iff Δ is conflict-free and defends all its elements; a *complete* assumption extension iff Δ is admissible and contains all assumptions it defends; a *preferred* (resp. *grounded*) assumption extension iff it is a (subset-)maximal (resp. (subset-)minimal) complete assumption extension; a *stable* assumption extension iff it is conflict-free and attacks every $\psi \in \mathcal{A} \setminus \Delta$; an *ideal* assumption extension iff it is a (subset-)maximal complete assumption extension that is contained in each preferred assumption extension.

There is a one-to-one correspondence between σ *argument extensions* and σ *assumption extensions*.

Let $claim(Ag)$ be the conclusion (or claim) of an argument Ag. The *conclusion* of a set of arguments $\mathcal{E}$ is defined as $\text{Concs}(\mathcal{E}) = \{c \in \mathcal{L} \mid c = claim(Ag)$ for an argument $Ag \in \mathcal{E}\}$.

Rationality postulates [6] are stated in terms of ABA in [12] as follows.

Definition 2 (**Rationality postulates** [12]). Let $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ be a flat ABA framework, where $\overline{}$ has the property such that contraries of assumptions are not assumptions. 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 $CN_{\mathcal{R}} : 2^{\mathcal{L}} \to 2^{\mathcal{L}}$ 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 $\mathcal{F}$, $\text{Concs}(\mathcal{E})$ is consistent (resp. $\text{Concs}(\mathcal{E})$ is closed).

We say that a set of arguments $\mathcal{E}$ is consistent if $\text{Concs}(\mathcal{E})$ is consistent [33].

In [6], rationality postulates are defined using notions such as *direct consistency*, *indirect consistency* and *closure-property*. If we use the notions, it may be said that under σ argumentation semantics, $AF_{\mathcal{F}}$ satisfies *closure* iff for each σ-extension $\mathcal{E}$ of $AF_{\mathcal{F}}$, $\text{Concs}(\mathcal{E}) = CN_{\mathcal{R}}(\text{Concs}(\mathcal{E}))$; $AF_{\mathcal{F}}$ satisfies *direct consistency* iff for each σ-extension $\mathcal{E}$, $\text{Concs}(\mathcal{E})$ is not contradictory; and $AF_{\mathcal{F}}$ satisfies *indirect consistency* iff for each σ-extension $\mathcal{E}$, $\text{Concs}(\mathcal{E})$ is consistent, that is, $CN_{\mathcal{R}}(\text{Concs}(\mathcal{E}))$ is not contradictory.

Heyninck and Arieli [18] proposed (generalized) assumption-based frameworks as follows. Let $\mathcal{L}$ be a propositional language, $p, p_i \in \mathcal{L}$ be atomic formulas (or atoms, for short), $\psi, \phi \in \mathcal{L}$ be compound formulas and $\Gamma, \Gamma', \Lambda \subseteq \mathcal{L}$ be sets of formulas. $\mathcal{L}$ contains conjunction "," disjunction "$\vee$", implication "$\rightarrow$", a negation operator "$\sim$", and a propositional constant T for truth.

A (propositional) *logic* for a language $\mathcal{L}$ [17,18] is a pair $\mathfrak{L} = (\mathcal{L}, \Vdash)$, where $\Vdash$ is a (Tarskian) consequence relation for $\mathcal{L}$ [36], which is a binary relation between sets of formulas and formulas in $\mathcal{L}$ satisfying the conditions such that it is reflexive (if $\psi \in \Gamma$ then $\Gamma \Vdash \psi$), monotonic (if $\Gamma \Vdash \psi$ and $\Gamma \subseteq \Gamma'$, then $\Gamma' \Vdash \psi$), and transitive (if $\Gamma \Vdash \psi$ and $\Gamma' \cup \{\psi\} \Vdash \phi$, then $\Gamma \cup \Gamma' \Vdash \phi$). The $\mathfrak{L}$-based transitive closure of a set Γ of $\mathcal{L}$-formulas is $\mathsf{Cn}_{\mathfrak{L}}(\Gamma) = \{\psi \in \mathcal{L} \mid \Gamma \Vdash \psi\}$ [17]. Let $\wp(\cdot)$ be the powerset operator.

Definition 3 (**Assumption-based frameworks** [18]). An *assumption-based framework* is a tuple $\mathbf{ABF} = \langle \mathfrak{L}, \Gamma, \Lambda, - \rangle$, where $\mathfrak{L} = \langle \mathcal{L}, \Vdash \rangle$ is a (propositional) Tarskian logic for a propositional language $\mathcal{L}$, Γ (the strict assumptions) and Λ (the candidate or defeasible assumptions) where $\Lambda \neq \{\}$ are distinct countable sets of $\mathcal{L}$-formulas, and $-{:}\Lambda \rightarrow \wp(\mathcal{L})$ is a *contrariness operator*, assigning a finite set of $\mathcal{L}$-formulas to every defeasible assumption in Λ.

In **ABF**, *attacks* is defined as follows: For $\Delta, \Theta \subseteq \Lambda$ and $\psi \in \Lambda$, Δ *attacks* ψ iff $\Gamma \cup \Delta \Vdash \phi$ for some $\phi \in -\psi$. Accordingly, Δ *attacks* Θ if Δ *attacks* some $\psi \in \Theta$.

The usual semantics in AA frameworks [8] is adapted to their ABFs as follows.

Definition 4 (**ABF semantics** [5]). Let $\mathbf{ABF} = \langle \mathfrak{L}, \Gamma, \Lambda, - \rangle$ be an assumption-based framework and let $\Delta \subseteq \Lambda$. A set of defeasible assumptions Δ is *closed* iff $\Delta = \{\alpha \in \Lambda \mid \Gamma \cup \Delta \Vdash \alpha\}$. **ABF** is said to be *flat* iff every set of defeasible assumptions Δ is closed. Then Δ is *conflict-free* iff there is no $\Delta' \subseteq \Delta$ that attacks some $\psi \in \Delta$. Δ is *stable* iff it is closed, conflict-free and attacks every $\psi \in \Lambda \setminus \Delta$.

2.2. Disjunctive logic programs

A disjunctive logic program (DLP) [26] is a finite set of rules of the form[4]

$$p_1 \vee \cdots \vee p_k \leftarrow p_{k+1}, \ldots, p_m, not\ p_{m+1}, \ldots, not\ p_n, \tag{1}$$

where $n \geqslant m \geqslant k > 0$. Each p_i ($1 \leqslant i \leqslant n$) is a ground atom. The symbol *not*[5] is the negation-as-failure (NAF) operator. An atom preceded by *not* is called a NAF-atom. Let P be a DLP and HB_P be the Herbrand base of P, i.e. the (finite) set of all ground atoms in the language of P. Let $M \subseteq HB_P$. A set M satisfies a ground rule of the form (1) if $\{p_{k+1}, \ldots, p_m\} \subseteq M$ and $\{p_{m+1}, \ldots, p_n\} \cap M = \emptyset$ imply $\{p_1, \ldots, p_k\} \cap M \neq \emptyset$. M is a *model* of P if it satisfies every ground rule in P. The *reduct* of P w.r.t. M is the DLP $P^M = \{p_1 \vee \ldots \vee p_k \leftarrow p_{k+1}, \ldots, p_m \mid$ there is a rule of the form $p_1 \vee \ldots \vee p_k \leftarrow p_{k+1}, \ldots, p_m, not\ p_{m+1}, \ldots, not\ p_n$ in P s.t. $\{p_{m+1}, \ldots, p_n\} \cap M = \emptyset\}$. Then M is a *stable model* of P if it is a $\subseteq$-minimal model of P^M [14,26].

In [18], stable models of a DLP P are defined based on $\mathcal{A}(P) \subseteq HB_P$, where $\mathcal{A}(P)$ is the set of atomic formulas that appear in P, and the symbol $\sim$ is used instead of *not* to denote NAF. Heyninck and Arieli [18] denoted by $\mathfrak{L} = \langle \mathcal{L}_{\mathrm{DLP}}, \Vdash \rangle$ the logic for the language $\mathcal{L}_{\mathrm{DLP}}$ consisting of disjunctions of atoms ($p_1 \vee \cdots \vee p_n$, for $n \geqslant 1$), NAF-atoms ($not\ p$) and rules of the form (1) of a DLP, where $\Vdash$ is constructed for $\mathcal{L}_{\mathrm{DLP}}$ by three inference rules: Modus Ponens (MP), Resolution (Res) and Reasoning by Cases (RBC). (The precise form of these inference rules will be described later in Remark 1.)

The ABF induced by a DLP P is defined by $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \sim\mathcal{A}(P), - \rangle$ [18], where $\mathfrak{L} = \langle \mathcal{L}_{\mathrm{DLP}}, \Vdash \rangle$, $\sim\mathcal{A}(P) = \{not\ p \mid p \in \mathcal{A}(P)\}$, and $-not\ p = \{p\}$ for every $p \in \mathcal{A}(P)$. All the ABFs induced by DLPs are based on the same core logic $\mathfrak{L} = \langle \mathcal{L}_{\mathrm{DLP}}, \Vdash \rangle$. Heyninck and Arieli showed a one-to-one correspondence between stable models of a DLP P and stable *assumption* extensions of $\mathbf{ABF}(P)$ as follows.

[4]A disjunctive logic program (DLP) defined in this paper is different from a normal disjunctive logic program (NDP) defined in Section 2.3. This is because we later consider transformation to *disjunctive default theories*, in which not a DLP but an NDP can be transformed to a disjunctive default theory. Details about this are discussed in Example 4.

[5]Gelfond and Lifschitz [15] use the symbol *not* to denote the negation-as-failure (NAF), while Przymusinski [26] uses the symbol $\sim$ to denote NAF. This paper uses Gelfond and Lifschitz's notation due to the purpose of this study.

Proposition 1 ([18, Proposition 2 and Proposition 3])**.** *Let P be a finite DLP and $M \subseteq \mathcal{A}(P)$. If M is a stable model of P, then $\overline{M} = \{not\ p \mid p \in (\mathcal{A}(P) \setminus M)\}$ is a stable (assumption) extension of $\boldsymbol{ABF}(P)$. Conversely if $\mathcal{E}$ is a stable (assumption) extension of $\boldsymbol{ABF}(P)$, then $\underline{\mathcal{E}} = \{p \in \mathcal{A}(P) \mid not\ p \notin \mathcal{E}\}$ is a stable model of P.*

2.3. *Extended disjunctive logic programs*

We consider a finite propositional extended disjunctive logic program (EDLP) [15] in this paper. An EDLP is a finite set of rules of the form:

$$L_1 | \dots | L_k \leftarrow L_{k+1}, \dots, L_m, not\ L_{m+1}, \dots, not\ L_n, \tag{2}$$

where $n \geqslant m \geqslant k > 0$. Each L_i $(1 \leqslant i \leqslant n)$ is a ground literal, that is, either a ground atom A (i.e. a propositional variable) or $\neg A$ preceded by classical negation "$\neg$", where $\neg A$ is called a *negative literal*. *not* denotes the negation-as-failure (NAF) as before, and *not* L is called a *NAF-literal*. The left (resp. right) part of $\leftarrow$ is the *head* (resp. the *body*). $body(r)$ denotes the body of a rule r from an EDLP. The symbol "|" is used to distinguish disjunction in the head of a rule from disjunction "$\vee$" used in classical logic. An EDLP is called a *normal disjunctive logic program* (NDP) if $\neg$ does not occur in it, while an EDLP is called an *extended logic program* (ELP) if it contains no disjunction ($k = 1$). An ELP is called a *normal logic program* (NLP) if $\neg$ does not occur in it. Let Lit_P (resp. HB_P) be the set of all ground literals (resp. all ground atoms) in the language of an EDLP P. When an EDLP P is an NDP (or NLP), Lit_P reduces to HB_P.

The semantics of an EDLP is given by *answer sets* [15].

Definition 5 (**answer sets**)**.** Let $S \subseteq Lit_P$. First, let P be a *not*-free EDLP (i.e. $m = n$ for each rule in P). Then, S is an *answer set* of P if S is a minimal set (w.r.t. $\subseteq$) satisfying the two conditions (i), (ii):

(i) For each rule $L_1 | \dots | L_k \leftarrow L_{k+1}, \dots, L_m$ in P, if $\{L_{k+1}, \dots, L_m\} \subseteq S$, then $L_i \in S$ for some i $(1 \leqslant i \leqslant k)$.
(ii) If S contains a pair of literals L and $\neg L$, then $S = Lit_P$.

Second, let P be any EDLP. The *reduct* P^S of P w.r.t. S is the *not*-free EDLP P^S such that $P^S = \{L_1 | \dots | L_k \leftarrow L_{k+1}, \dots, L_m \mid$ there is a rule of the form $L_1 | \dots | L_k \leftarrow L_{k+1}, \dots L_m, not\ L_{m+1}, \dots$ $not\ L_n$ in P s.t. $\{L_{m+1}, \dots, L_n\} \cap S = \emptyset\}$. Then S is an answer set of P if S is the answer set of P^S.

An answer set S is *consistent* if $S \neq Lit_P$; otherwise S is *contradictory*. An EDLP P having a consistent answer set is *consistent*; otherwise P is *inconsistent* under answer set semantics.

It is considered, for an answer set S and a literal L, (i) $S \models L$ iff $L \in S$, (ii) $S \models not\ L$ iff $L \notin S$. A literal $L \in Lit_P$ is interpreted "*unknown*" under the answer set semantics if $L \notin S$ and $\neg L \notin S$ for any answer set S of an EDLP P.

The following example illustrates the difference between "|" and "$\vee$" in logic programming.

Example 2. The EDLP $\{p \mid \neg p \leftarrow\}$ has two answer sets $\{p\}$ and $\{\neg p\}$. If we use $\vee$ instead of |, however, $\{p \vee \neg p \leftarrow\}$ is logically equivalent to $\{p \leftarrow p\}$ under classical logic, which has the minimal model (stable model) $\{\}$. Hence, | and $\vee$ have different meanings in general.

The semantics of an EDLP is also given by (four-valued) *paraconsistent stable models* (or *p-stable models*[6]) [28], which are regarded as answer sets defined without the condition (ii) in Definition 5, denoting that we don't get the trivialization or *deductive explosion* of inconsistent (p-stable) models.

Definition 6 (p-stable models). Let $M \subseteq Lit_P$. First, let P be a *not*-free EDLP (i.e. $m = n$ for each rule). Then, M is a *p-stable model* of P if M is a minimal set (w.r.t. $\subseteq$) satisfying the following condition: For each rule $L_1|\ldots|L_k \leftarrow L_{k+1}, \ldots, L_m$ in P, if $\{L_{k+1}, \ldots, L_m\} \subseteq M$, then $L_i \in M$ for some i ($1 \leqslant i \leqslant k$). Second, let P be any EDLP. The *reduct* P^M of P w.r.t. M is the *not*-free EDLP P^M defined in Definition 5. Then M is a p-stable model of P if M is a p-stable model of P^M.

In [28], $I \subseteq Lit_P$ is considered as an interpretation which is defined as a function $I : Lit_P \rightarrow \{\mathbf{t}, \mathbf{f}, \top, \bot\}$, where $\mathbf{t}, \mathbf{f}, \top, \bot$ are the four-valued truth values denoting *true*, *false*, *contradictory*, and *undefined*. Then, the truth value $I(L)$ is assigned to each literal $L \in Lit_P$ such that:

(i) $I(L) = \mathbf{t}$ if $L \in I$ and $\neg L \notin I$, (ii) $I(L) = \mathbf{f}$ if $L \notin I$ and $\neg L \in I$,
(iii) $I(L) = \top$ if both $L \in I$ and $\neg L \in I$, (iv) $I(L) = \bot$ otherwise.

A p-stable model M is *inconsistent* if M contains a pair of complementary literals (in other words, there exists a literal $L \in M$ s.t. $M(L) = \top$); otherwise M is *consistent*. An EDLP P is *consistent* if P has a consistent p-stable model; otherwise P is *inconsistent* under paraconsistent stable model semantics.

Example 3. Consider the EDLP $P = \{\neg p \leftarrow, p \leftarrow, q \leftarrow\}$. P has the inconsistent answer set Lit_P which is contradictory both w.r.t. p and q. P has also the inconsistent p-stable model $M = \{\neg p, p, q\}$, where contradiction is localized w.r.t. p, and the consistent information about q is obtained (i.e. $M(p) = \top$, $M(q) = \mathbf{t}$). This shows that a p-stable model is *paraconsistent* even if it is inconsistent.

Gelfond et al. [16] proposed a disjunctive default theory (*ddt*, for short), which is a set of disjunctive defaults whose form is $\frac{\alpha:\, \beta_1,\ldots,\beta_m}{\gamma_1|\gamma_2|\ldots|\gamma_n}$. The semantics of a *ddt* is given by extensions which are generalization of Reiter's extensions for a default theory [27]. Regarding logic programming, they showed in the following theorem that a propositional EDLP P can be translated into a disjunctive default theory $emb(P)$ by replacing every rule in P of the form (2) with the disjunctive default

$$\frac{L_{k+1} \wedge \ldots \wedge L_m : \neg L_{m+1}, \ldots, \neg L_n}{L_1|\ldots|L_k}.$$

The intuition behind the disjunctive default can be "if each of $L_{k+1}, \ldots, L_m$ is believed and if each of $\neg L_{m+1}, \ldots, \neg L_n$ can be consistently believed, then L_i is believed for some i ($1 \leqslant i \leqslant k$)".

The disjunctive defaults of this form are used to compute extensions of the *ddt* $emb(P)$. There exists the correspondence between answer sets of an EDLP P and extensions of $emb(P)$ as follows.

Theorem 1 ([16, Theorem 7.2])**.** *Let P be a propositional EDLP. Then S is an answer set of P iff S is the set of all literals from an extension of the disjunctive default theory $emb(P)$.*

Let $\mathcal{F}(P) = \langle \mathcal{L}_P, P, \mathcal{A}_P, \overline{} \rangle$ be the ABA framework translated from an ELP P, where $NAF_P = \{not\ L|\ L \in Lit_P\}$, $\mathcal{L}_P = Lit_P \cup NAF_P$, $\mathcal{A}_P = NAF_P$, and $\overline{not\ L} = L$ for each $not\ L \in \mathcal{A}_P$. Given $M \subseteq Lit_P$, let $\Delta_M = \{not\ L \mid L \in Lit_P \setminus M\}$.[7] Let $P_{tr} \stackrel{\text{def}}{=} P \cup \{L \leftarrow p, \neg p \mid p \in Lit_P,\ L \in Lit_P\}$ be

[6] In this paper, the term "p-stable models" is used not as an abbreviation of *partial* stable model semantics by Przymusinski [26] but as that of *paraconsistent* stable model semantics by Sakama and Inoue [28]. In [28], a p-stable model is defined for an EDLP whose rule head uses $\vee$ rather than $|$. However notice that, a p-stable model is defined regardless the connective of disjunction used in rule head (e.g. "$\vee$", "$|$").

the ELP obtained from an ELP P by incorporating the *trivialization rules* [28]. It was shown that answer sets (resp. paraconsistent stable models) of an ELP P can be captured by stable *argument* extensions of the ABA framework translated from the ELP P_{tr} (resp. P) as follows.

Theorem 2 ([32, Theorem 3]). *Let P be an ELP and $M \subseteq Lit_P$. Then M is a p-stable model of P iff there is a stable (argument) extension $\mathcal{E}$ of the ABA framework $\mathcal{F}(P)$ such that $M \cup \Delta_M = \mathrm{Concs}(\mathcal{E})$ (in other words, $M = \mathrm{Concs}(\mathcal{E}) \cap Lit_P$).*

Theorem 3 ([32, Theorem 4]). *Let P be an ELP and $S \subseteq Lit_P$. Then S is an answer set of P iff there is a stable (argument) extension $\mathcal{E}_{tr}$ of the ABA framework $\mathcal{F}(P_{tr})$ such that $S \cup \Delta_S = \mathrm{Concs}(\mathcal{E}_{tr})$ (in other words, $S = \mathrm{Concs}(\mathcal{E}_{tr}) \cap Lit_P$).*

For a consistent ELP, the following theorem holds.

Theorem 4 ([33, Theorem 5]). *Let P be a consistent ELP and $S \subseteq Lit_P$. Then S is an answer set of P iff there is a consistent stable (argument) extension $\mathcal{E}$ of the ABA framework $\mathcal{F}(P)$ such that $S \cup \Delta_S = \mathrm{Concs}(\mathcal{E})$.*

Notice that Δ_M in Theorem 2 (resp. Δ_S in Theorem 3) is a stable *assumption* extension of the ABA framework $\mathcal{F}(P)$ (resp. $\mathcal{F}(P_{tr})$), while Δ_S in Theorem 4 is a *consistent* stable assumption extension of the ABF $\mathcal{F}(P)$ due to Proposition 12, Proposition 13 and Proposition 14 as proved in Appendix respectively.

3. ABA for extended disjunctive logic programming

3.1. Assumption-based frameworks translated from EDLPs

We propose an assumption-based framework (ABF) translated from an EDLP, which incorporates explicit negation along with | instead of $\vee$ in Heyninck and Arieli's ABF induced by a DLP to achieve our purpose shown in the introduction. An ABF translated from an EDLP is based on the logic constructed by three inference rules: Modus Ponens (MP), Resolution (Res) and Reasoning by Cases (RBC):

$$[\mathsf{MP}] \quad \frac{\psi \leftarrow \phi_1, \ldots, \phi_n \qquad \phi_1 \quad \phi_2 \quad \cdots \quad \phi_n}{\psi}$$

$$[\mathsf{Res}] \quad \frac{\psi_1'|\ldots|\psi_m'|\ell_1|\ldots|\ell_n|\psi_1''|\ldots|\psi_k'' \qquad not\ \ell_1 \quad \cdots \quad not\ \ell_n}{\psi_1'|\ldots|\psi_m'|\psi_1''|\ldots|\psi_k''}$$

$$[\mathsf{RBC}] \quad \frac{\begin{array}{cccccc} \ell_1 & \ell_2 & & \ell_n & & \\ \vdots & \vdots & & \vdots & & \\ \psi & \psi & \cdots & \psi & & \ell_1|\ldots|\ell_n \end{array}}{\psi}$$

where | is the connective of a disjunction, ℓ_i is a propositional literal, each $\phi_i \in \{\ell_i, not\ \ell_i\}$ is a propositional literal or its NAF-literal, and ψ, ψ_i are disjunctions of propositional literals using |.

[7] In [32], $\neg.CM$ is used rather than Δ_M to refer to the set $\{not\ L \mid L \in Lit_P \setminus M\}$. However for notational convenience, Δ_M is used instead of $\neg.CM$ to refer to this set in this paper.

$\psi \leftarrow \phi_1, \ldots, \phi_n$ is a rule of the form (2). Since $\psi \leftarrow$ is identified with $\psi \leftarrow \mathsf{T}$, [MP] implies Reflexivity: [Ref] $\frac{\psi \leftarrow}{\psi}$.

Definition 7. We denote by $\mathfrak{L} = \langle \mathcal{L}_{\mathrm{EDLP}}, \Vdash \rangle$ the logic for the language $\mathcal{L}_{\mathrm{EDLP}}$ which consists of disjunctions of propositional literals ($\ell_1 | \ldots | \ell_n$, for $n \geqslant 1$), NAF-literals (*not* ℓ) and rules of the form (2) of an EDLP, where $\Vdash$ is constructed for $\mathcal{L}_{\mathrm{EDLP}}$ by three inference rules: Modus Ponens (MP), Resolution (Res) and Reasoning by Cases (RBC) above. In other words, $\Vdash$ denotes derivability using three inference rules: [MP] (including [Ref]), [Res] and [RBC].

Remark 1. Heyninck and Arieli's ABF [18] is based on the logic $\mathfrak{L} = \langle \mathcal{L}_{\mathrm{DLP}}, \Vdash \rangle$, where $\Vdash$ is constructed for $\mathcal{L}_{\mathrm{DLP}}$ by three inference rules: [MP], [Res] and [RBC] having the restricted forms such that $|$ (resp. a literal ℓ_i) is replaced with $\vee$ (resp. an atom p_i), each $\phi_i \in \{p_i, not\ p_i\}$ is an atom or a NAF-atom, and ψ, ψ_i are disjunctions of atoms ($p_1 \vee \cdots \vee p_n$, for $n \geqslant 1$).

$\Sigma \Vdash \varphi$ holds iff φ is either in Σ or is derived from Σ using the three inference rules above. According to Definition 3, $\Sigma \Vdash \varphi$ iff $\varphi \in \mathsf{Cn}_{\mathfrak{L}}(\Sigma)$, where $\mathsf{Cn}_{\mathfrak{L}}(\Sigma)$ is the $\mathfrak{L}$-based transitive closure of Σ (namely, the $\subseteq$-smallest set that contains Σ and is closed under [MP], [Res] and [RBC]). Notice that for any $\varphi \in \mathsf{Cn}_{\mathfrak{L}}(\Sigma)$, if φ is not of the form $\ell_1 | \ldots | \ell_n$, then $\varphi \in \Sigma$.

For a special ABF $= \langle \mathfrak{L}, \Gamma, \Lambda, - \rangle$ such that each defeasible assumption from Λ has a unique contrary (i.e. $|-\alpha| = 1$[8] for $\forall \alpha \in \Lambda$), we may denote such an ABF by a tuple $\langle \mathfrak{L}, \Gamma, \Lambda, \overline{\ } \rangle$, where $\overline{\ }$ is a total mapping from Λ into $\mathcal{L}$, and $\overline{\alpha} \in \mathcal{L}$ is the *contrary* of $\alpha \in \Lambda$.

In what follows, let $NAF_P = \{not\ \ell \mid \ell \in Lit_P\}$ and $\mathbb{L}_P = Lit_P \cup NAF_P$ for an EDLP P. We are now ready to define an ABF translated from an EDLP.

Definition 8. Let P be an EDLP. The assumption-based framework (ABF) translated from P is defined by $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{\ } \rangle$, where $\mathfrak{L} = \langle \mathcal{L}_{\mathrm{EDLP}}, \Vdash \rangle$, $\mathcal{A}_P = NAF_P = \{not\ \ell \mid \ell \in Lit_P\}$, and $\overline{not\ \ell} = \ell$ for every $not\ \ell \in \mathcal{A}_P$.

In the following, we show the ABF translated from an EDLP P is always *flat*.

Proposition 2. *$\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{\ } \rangle$ translated from an EDLP P is flat.*

Proof. Let $\Delta \subseteq \mathcal{A}_P$. Since $\Vdash$ is reflexive, it holds that (i) $P \cup \Delta \Vdash \alpha$ for $\forall \alpha \in \Delta$. On the other hand, it holds that (ii) $P \cup \Delta \nVdash \beta$ for $\forall \beta \in \mathcal{A}_P \setminus \Delta$, since NAF-literals, which is the only form the defeasible assumptions in $\mathbf{ABF}(P)$ can take, do not occur in the heads of rules from P. Then due to (i),(ii), it holds that $\Delta = \{\alpha \in \mathcal{A}_P \mid P \cup \Delta \Vdash \alpha\}$ for $\forall \Delta \subseteq \mathcal{A}_P$. Thus $\mathbf{ABF}(P)$ is flat. $\square$

In $\mathbf{ABF}(P)$, the semantics is given by *assumption* extensions as follows.

Definition 9. Let $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{\ } \rangle$ be the ABF translated from an EDLP P. Let $\Delta \subseteq \mathcal{A}_P$ and $\alpha \in \mathcal{A}_P$. Δ is *conflict-free* iff there is no $\Delta' \subseteq \Delta$ that attacks some $\psi \in \Delta$. Δ *defends* α iff each Δ' that attacks α is attacked by Δ. Then Δ is: *admissible* iff Δ is conflict-free and defends all its elements; a *complete* assumption extension iff Δ is admissible and contains all assumptions it defends; a *preferred* (resp. *grounded*) assumption extension iff it is a (subset-)maximal (resp. (subset-)minimal) complete assumption extension; a *stable* assumption extension iff it is is conflict-free and attacks every

[8]For a set S, $|S|$ denotes the cardinality of S.

$\psi \in \mathcal{A}_P \setminus \Delta$; an *ideal* assumption extension iff it is a (subset-)maximal complete assumption extension that is contained in each preferred assumption extension.

In **ABF**(P), *consistency* of a set of literals and NAF-literals $X \subseteq \mathbb{L}_P$ is defined using a consequence operator CN_P which is adapted from the $CN_{\mathcal{R}}$ operator found in standard ABA.

Definition 10. Let $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{\ \ }\rangle$ be the ABF translated from an EDLP P, where $\mathfrak{L} = \langle \mathcal{L}_{\text{EDLP}}, \Vdash\rangle$. For a set $X \subseteq \mathbb{L}_P = Lit_P \cup NAF_P$, X is said to be *contradictory* iff X is contradictory w.r.t. $\overline{\ \ }$, i.e. there exists an assumption $\alpha \in \mathcal{A}_P$ s.t. $\{\alpha, \overline{\alpha}\} \subseteq X$; or X is contradictory w.r.t. $\neg$, i.e. there exists $s \in \mathbb{L}_P$ s.t. $\{s, \neg s\} \subseteq X$. Let $\mathsf{CN}_P : \wp(\mathbb{L}_P) \to \wp(\mathbb{L}_P)$ be a consequence operator such that for $X \subseteq \mathbb{L}_P$,

$$\mathsf{CN}_P(X) \stackrel{\text{def}}{=} \{\phi \in \mathbb{L}_P \mid P \cup X \Vdash \phi\} = \mathsf{Cn}_{\mathfrak{L}}(P \cup X) \cap \mathbb{L}_P.$$

$\mathsf{CN}_P(X)$ is said to be the closure of X. X is said to be *closed* w.r.t. CN_P iff $X = \mathsf{CN}_P(X)$. A set $X \subseteq \mathbb{L}_P$ is said to be *inconsistent* iff the closure $\mathsf{CN}_P(X)$ is contradictory. X is said to be *consistent* iff it is not inconsistent.

3.2. Correspondence between answer sets of an EDLP and stable assumption extensions

Proposition 1 for a DLP [18] is generalized to Proposition 4 and Proposition 5 for an EDLP shown below. To this end, firstly as the extension of Proposition 1, we prepare the following corollary for a DLP whose stable models are defined based on HB_P.

Corollary 1. *The extended ABF induced by a DLP P is defined by $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \sim HB_P, -\rangle$, where $\mathfrak{L} = \langle \mathcal{L}_{\text{DLP}}, \Vdash\rangle$, $\sim HB_P = \{not\ p \mid p \in HB_P\}$ and $-not\ p = \{p\}$ for every $p \in HB_P$. Let $M \subseteq HB_P$. (i) If M is a stable model of P, then $\Delta = \{not\ p \mid p \in (HB_P \setminus M)\}$ is a stable assumption extension of $\mathbf{ABF}(P)$. (ii) Conversely if Δ is a stable assumption extension of $\mathbf{ABF}(P)$, then $M = \{p \in HB_P \mid not\ p \notin \Delta\}$ is a stable model of P.*

Proof. Let $\Delta_{\mathcal{A}(P)} = \{not\ p \mid p \in (HB_P \setminus \mathcal{A}(P))\}$.

(i) Let M be a stable model of P, where $M \subseteq \mathcal{A}(P) \subseteq HB_P$. Then for $\overline{M} = \{not\ p \mid p \in (\mathcal{A}(P) \setminus M)\}$, it holds that $\overline{M} \cup \Delta_{\mathcal{A}(P)} = \Delta = \{not\ p \mid p \in (HP_P \setminus M)\}$. Based on [18, Corollary 1], it holds that,

$$\text{for } p \in \mathcal{A}(P),\ p \in M \ \text{ iff }\ p \in \mathsf{Cn}_{\mathfrak{L}}(P \cup \overline{M}) \ \text{ iff } p \in \mathsf{Cn}_{\mathfrak{L}}(P \cup \Delta). \quad \text{Hence,}$$

$$\text{for } p \in HB_P,\ p \in M \ \text{ iff }\ p \in \mathsf{Cn}_{\mathfrak{L}}(P \cup \Delta) \ \text{ iff }\ P \cup \Delta \Vdash p.$$

Thus Δ is conflict-free and attacks every $not\ p \notin \Delta$ for $p \in HB_P$, which means that Δ is a stable assumption extension of $\mathbf{ABF}(P)$.

(ii) Conversely, let $\Delta \subseteq HB_P$ be a stable assumption extension of $\mathbf{ABF}(P)$. Then for $\forall \alpha \in \{not p \notin \Delta \mid p \in HB_P\}$, Δ attacks α. This means that, $P \cup \Delta \Vdash p$, that is, $p \in \mathsf{Cn}_{\mathfrak{L}}(P \cup \Delta)$ for $not\ p \notin \Delta$ where $p \in HB_P$.

On the other hand, based on [18, Corollary 1], it holds that, for a stable model M of P,

$$\begin{aligned} & p \in M \quad \text{iff} \quad p \in \mathsf{Cn}_{\mathfrak{L}}(P \cup \overline{M}) \quad \text{for } p \in \mathcal{A}(P) \\ & \text{iff} \quad p \in \mathsf{Cn}_{\mathfrak{L}}(P \cup \overline{M} \cup \Delta_{\mathcal{A}(P)}) = \mathsf{Cn}_{\mathfrak{L}}(P \cup \Delta) \quad \text{for } p \in HB_P. \end{aligned} \tag{3}$$

Thus it holds that, $p \in M$ for $not\ p \notin \Delta$ and $p \in HB_P$ where Δ is a stable assumption extension. $\square$

Next, Corollary 1 for a DLP is mapped to Proposition 3 for an NDP based on two lemmas about DLPs and NDPs as follows.

Lemma 1. *Let P be a propositional NDP and P_D be the DLP translated from P which is obtained by replacing a rule: $p_1|\ldots|p_k \leftarrow p_{k+1}, \ldots, p_m, not\ p_{m+1}, \ldots, not\ p_n$ from P with the rule $p_1 \vee \ldots \vee p_k \leftarrow p_{k+1}, \ldots, p_m, not\ p_{m+1}, \ldots, not\ p_n$, where $HB_{P_D} = HB_P$. Let $M \subseteq HB_P$. Then M is an answer set of P iff M is a stable model of P_D.*

Proof. Given $M \subseteq HB_P$, it holds that, $\{p_1, \ldots, p_k\} \cap M \neq \emptyset$ iff $p_i \in M$ for some i $(1 \leqslant i \leqslant k)$. This is used in the following proof.

$\Rightarrow$: Suppose that $M \subseteq HB_P$ is an answer set of P. First, let P be a *not*-free NDP (i.e., $m = n$ in (2)). Since M is an answer set of P, M is a $\subseteq$-minimal set satisfying the condition such that if $\{p_{k+1}, \ldots, p_m\} \subseteq M$, then some $p_i \in M$ $(1 \leqslant i \leqslant k)$ for each rule $p_1|\ldots|p_k \leftarrow p_{k+1}, \ldots, p_m$ in P. Hence M is a $\subseteq$-minimal set satisfying the condition such that if $\{p_{k+1}, \ldots, p_m\} \subseteq M$, then $\{p_1, \ldots, p_k\} \cap M \neq \emptyset$ for each rule $p_1 \vee \ldots \vee p_k \leftarrow p_{k+1}, \ldots, p_m$ in P_D, which means that M is a $\subseteq$-minimal model of P_D. Now since $(P_D)^M = P_D$ holds for the *not*-free P_D, M is a stable model of P_D. Second, let P be any NDP. Since M is an answer set of P, M is an answer set of the *not*-free NDP P^M. Therefore, M is a stable model of the *not*-free DLP $(P_D)^M$, which means that M is a $\subseteq$-minimal model of the DLP $((P_D)^M)^M = (P_D)^M$. Hence M is a stable model of P_D.

$\Leftarrow$: Suppose that $M \subseteq HB_P$ is a stable model of P_D. First, let P_D be the *not*-free DLP (i.e., $m = n$ in (1)). Since M is a stable model of the *not*-free P_D, M is a $\subseteq$-minimal model of $(P_D)^M = P_D$ which satisfies the condition such that if $\{p_{k+1}, \ldots, p_m\} \subseteq M$, then $\{p_1, \ldots, p_k\} \cap M \neq \emptyset$ for each rule $p_1 \vee \ldots \vee p_k \leftarrow p_{k+1}, \ldots, p_m$ in $(P_D)^M = P_D$. Therefore, M is a $\subseteq$-minimal set of P satisfying the condition such that if $\{p_{k+1}, \ldots, p_m\} \subseteq M$, then some $p_i \in M$ $(1 \leqslant i \leqslant k)$ for each rule $p_1|\ldots|p_k \leftarrow p_{k+1}, \ldots, p_m$ in P, which means that M is an answer set of the *not*-free NDP P. Second, let P_D be any DLP. Since M is a stable model of P_D, M is a $\subseteq$-minimal model of the DLP $(P_D)^M$ which is *not*-free. Hence M is a $\subseteq$-minimal set satisfying the condition such that if $\{p_{k+1}, \ldots, p_m\} \subseteq M$, then $\{p_1, \ldots, p_k\} \cap M \neq \emptyset$ for each rule $p_1 \vee \ldots \vee p_k \leftarrow p_{k+1}, \ldots, p_m$ in $(P_D)^M$, which means that, M is a $\subseteq$-minimal set satisfying the condition such that if $\{p_{k+1}, \ldots, p_m\} \subseteq M$, then some $p_i \in M$ $(1 \leqslant i \leqslant k)$ for each rule $p_1|\ldots|p_k \leftarrow p_{k+1}, \ldots, p_m$ in P^M. Therefore M is an answer set of P^M. Hence M is an answer set of P. $\square$

Though Lemma 1 holds, the difference between NDPs and DLPs appears when considering their relation to (disjunctive) default logic as shown in the following example. This implies that an NDP can be transformed to a disjunctive default theory but a DLP can never be done so.

Example 4. Heyninck and Arieli considered the DLP $\pi_1 = \{p \vee q \leftarrow,\ \ q \leftarrow p,\ \ p \leftarrow q\}$ in [18, Example 1], which has the unique stable model $M = \{p, q\}$. By replacing $\vee$ with $|$ in π_1, we obtain the NDP $\pi_2 = \{p \mid q \leftarrow,\ \ q \leftarrow p,\ \ p \leftarrow q\}$ which has the unique answer set $S = \{p, q\}$. Then thanks to Gelfond et al.'s Theorem 1 [16, Theorem 7.2], π_2 can be translated into the disjunctive default theory d_2 (namely, $emb(\pi_2)$) below. Instead, π_1 is expressed by the standard default theory d_1 as follows since it uses the connective $\vee$ rather than $|$ in rule head.

$$d_1 = \left\{\frac{p:}{q}, \frac{q:}{p},\ p \vee q\right\}, \qquad d_2 = \left\{\frac{p:}{q}, \frac{q:}{p},\ p \mid q\right\}.$$

As a result, d_1 has the unique extension E_1 consisting of $p \vee q$ and its logical consequences based on Reiter's default theory [27], while d_2 has the unique extension E_2 consisting of p, q and their logical consequences based on Gelfond et al.'s disjunctive default theory [16]. Therefore there is no relationship between E_1 and M, while there is the correspondence between E_2 and S such that $S = E_2 \cap Lit_{\pi_2}$. Thus the default theory d_1 corresponding to the DLP π_1 has the same difficulty as D_1 addressed in Example 1, while the disjunctive default theory d_2 corresponding to the NDP π_2 avoids such a difficulty.

The following lemma is also needed to obtain Proposition 3 for an NDP.

Lemma 2. *Let $M \subseteq HB_P$ be an answer set of an NDP P and let $\Delta_M = \{not\ L \mid L \in HB_P \setminus M\}$. Then $p \in M$ iff $p \in \mathsf{Cn}_{\mathfrak{L}}(P \cup \Delta_M)$ for $p \in HB_P$.*

Proof. Let P_D be the DLP translated from an NDP P as defined in Lemma 1.

$p \in M$ iff $p \in \mathsf{Cn}_{\mathfrak{L}}(P_D \cup \Delta_M)$ for $p \in HB_P$, where $\mathfrak{L} = \langle \mathcal{L}_{\text{DLP}}, \Vdash \rangle$, (due to Lemma 1 and (3))

iff $p \in \mathsf{Cn}_{\mathfrak{L}}(P \cup \Delta_M)$ for $p \in HB_P$, where $\mathfrak{L} = \langle \mathcal{L}_{\text{EDLP}}, \Vdash \rangle$. □

Proposition 3. *Let P be an NDP, $M \subseteq HB_P$, and $\boldsymbol{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$, where $\mathfrak{L} = \langle \mathcal{L}_{\text{EDLP}}, \Vdash \rangle$. (i) If M is an answer set of P, then $\Delta = \{not\ p \mid p \in (HB_P \setminus M)\}$ is a stable assumption extension of $\boldsymbol{ABF}(P)$. (ii) Conversely if Δ is a stable assumption extension of $\boldsymbol{ABF}(P)$, then $M = \{p \in HB_P \mid not\ p \notin \Delta\}$ is an answer set of P.*

Proof. This follows from Corollary 1 for a DLP based on Lemma 1 and Lemma 2. □

Now, we show that p-stable models (resp. answer sets) of an EDLP P are captured by stable assumption extensions of the ABF translated from P (resp. P_{tr}). To this end, the notation "+" [15,28] is used.

A *positive form* of an EDLP P [28] is the NDP P^+ which is obtained by replacing each negative literal $\neg L$ in P (where L is an atom) with a corresponding newly introduced atom L' (called the positive form of $\neg L$ [15]) in P^+. Let M^+ be an answer set of such P^+. Then the following lemma holds by definition.

Lemma 3. *Let P be an EDLP and P^+ be its positive form. Let $M \subseteq Lit_P$. Then M is a p-stable model iff M^+ is an answer set of P^+.*

Proposition 4. *Let $\boldsymbol{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$ be the ABF translated from an EDLP P and $M \subseteq Lit_P$.*

If M is a p-stable model of P, then $\Delta = \{not\ \ell \mid \ell \in (Lit_P \setminus M)\}$ is a stable assumption extension of $\boldsymbol{ABF}(P)$. Conversely if Δ is a stable assumption extension of $\boldsymbol{ABF}(P)$, then $M = \{\ell \in Lit_P \mid not\ \ell \notin \Delta\}$ is a p-stable model of P.

Proof. Let P^+ be the NDP which is the positive form of an EDLP P. For a set $S \subseteq Lit_P$, let S^+ be the set obtained by replacing each negative literal $\neg L$ in S with a newly introduced atom L'. Then the Herbrand base HB_{P^+} of P^+ is $(Lit_P)^+$. Moreover, if a set $U^+ \subseteq HB_{P^+}$ is given, we denote by U the set obtained by replacing each newly introduced atom $L' \in U^+$ with the corresponding original literal $\neg L \in Lit_P$.

(i) Let M be a p-stable model of an EDLP P. Then due to Lemma 3, M^+ is an answer set of the NDP P^+. Hence $\Delta_{M^+} = \{not\ p \mid p \in (HB_{P^+} \setminus M^+)\}$ is a stable assumption extension of $\mathbf{ABF}(P^+)$ due to Proposition 3. Thus $\Delta = \Delta_M = \{not\ \ell \mid \ell \in (Lit_P \setminus M)\}$ is a stable assumption extension of $\mathbf{ABF}(P)$.

(ii) Conversely, let $\Delta = \Delta_M = \{not\ \ell | \ell \in (Lit_P \setminus M)\}$ be a stable assumption extension of $\mathbf{ABF}(P)$. Then $\Delta_{M^+} = \{not\ p|\ p \in (HB_{P^+} \setminus M^+)\} = \{not\ p|p \in (Lit_P \setminus M)^+)\}$ is a stable assumption extension of $\mathbf{ABF}(P^+)$. Thus due to Proposition 3, $M^+ = \{p \in HB_{P^+}|not\ p \notin \Delta_{M^+}\}$ is the answer set of the NDP P^+. Hence $M = \{\ell \in Lit_P|not\ \ell \notin \Delta_M\}$ where $\Delta = \Delta_M$ is a p-stable model of P due to Lemma 3. □

Proposition 5. *Let P be an EDLP, $S \subseteq Lit_P$ and $\boldsymbol{ABF}(P_{tr}) = \langle \mathfrak{L}, P_{tr}, \mathcal{A}_P, \overline{\ \ }\rangle$ be the ABF translated from the EDLP $P_{tr} = P \cup \{L \leftarrow p, \neg p \mid p \in Lit_P,\ L \in Lit_P\}$, where $\mathbb{L}_{P_{tr}} = \mathbb{L}_P$. If S is an answer set of P, then $\Delta = \{not\ \ell \mid \ell \in (Lit_P \setminus S)\}$ is a stable assumption extension of $\boldsymbol{ABF}(P_{tr})$. Conversely if Δ is a stable assumption extension of $\boldsymbol{ABF}(P_{tr})$, then $S = \{\ell \in Lit_P \mid not\ \ell \notin \Delta\}$ is an answer set of P.*

Proof. [28, Theorem 3.5] shows that S is an answer set of an EDLP P iff S is p-stable model of P_{tr}. Then this follows from Proposition 4 based on [28, Theorem 3.5]. □

$\mathsf{CN}_P(\Delta)$ gives us the *conclusion* of an assumption extension Δ. In Proposition 6 below, we show the relationship between answer sets (or p-stable models) of an EDLP and the conclusions of assumption extensions of the translated ABF.

Lemma 4. *Let M be a p-stable model of an EDLP P and $\Delta_M = \{not\ \ell | \ \ell \in (Lit_P \setminus M)\}$. Then $\ell \in M$ iff $\ell \in \mathsf{Cn}_{\mathfrak{L}}(P \cup \Delta_M)$ for $\ell \in Lit_P$.*

Proof. This follows from Lemma 2 based on the renaming technique used in the proof of Proposition 4. □

Proposition 6. *Let P be an EDLP, M (resp. S) be a p-stable model (resp. an answer set) of P, and $\Delta_M = \{not\ \ell \mid \ell \in (Lit_P \setminus M)\}$ (resp. $\Delta_S = \{not\ \ell \mid \ell \in (Lit_P \setminus S)\}$) be the stable assumption extension of $\boldsymbol{ABF}(P)$ (resp. $\boldsymbol{ABF}(P_{tr})$). Then it holds that, $\mathsf{CN}_P(\Delta_M) = M \cup \Delta_M$, and $\mathsf{CN}_{P_{tr}}(\Delta_S) = S \cup \Delta_S$.*

Proof.

1. Let M be a p-stable model of an EDLP P. Then for $\ell \in Lit_P$, it holds that $\ell \in M$ iff $\ell \in \mathsf{Cn}_{\mathfrak{L}}(P \cup \Delta_M)$ iff $\ell \in \mathsf{CN}_P(\Delta_M)$ according to Lemma 4. Besides *not* $\ell \in \Delta_M$ iff *not* $\ell \in \mathsf{CN}_P(\Delta_M)$. Hence $\mathsf{CN}_P(\Delta_M) = M \cup \Delta_M$ holds.
2. Let S be an answer set of P. Then S is a p-stable model of P_{tr} due to [28, Theorem 3.5]. Besides $\Delta_S = \{not\ \ell \mid \ell \in (Lit_P \setminus S)\}$ is the stable assumption extension of $\mathbf{ABF}(P_{tr})$ due to Proposition 5. Then by applying the result of Item 1 to a p-stable model S of P_{tr}, we obtain $\mathsf{CN}_{P_{tr}}(\Delta_S) = S \cup \Delta_S$. □

In all examples shown in this paper, we assume that Lit_P i.e. the set of all ground literals in the language of an EDLP P coincides with the set $\{L \mid L$ or $\neg L$ appears in $P\}$[9] where $\neg\neg L = L$.

Example 5 (Cont. Example 1). Consider Kyoto protocol problem. For the EDLP P_1, $Lit_{P_1} = \{p, r, y, f, k, \neg p, \neg r, \neg y, \neg f, \neg k\}$, and $(P_1)_{tr} = P_1 \cup \{L \leftarrow p, \neg p \mid p \in Lit_{P_1},\ L \in Lit_{P_1}\}$. Answer sets of $(P_1)_{tr}$ coincide with those of P_1, that is, $S_1 = \{p, r, k\}$ and $S_2 = \{y, f, k\}$. According to

[9] This assumption is also used in all examples given in [32,33] though it is not stated there.

Proposition 5, $\mathbf{ABF}((P_1)_{tr})$ (resp. $\mathbf{ABF}(P_1)$) has two stable assumption extensions Δ_1 and Δ_2 such that

$$\Delta_1 = \{not\ y, not\ f, not\ \neg p, not\ \neg r, not\ \neg y, not\ \neg f, not\ \neg k\},$$
$$\Delta_2 = \{not\ p, not\ r, not\ \neg p, not\ \neg r, not\ \neg y, not\ \neg f, not\ \neg k\}.$$

The conclusion of each assumption extension Δ_i defined as $\mathsf{CN}_{P_1}(\Delta_i)$ $(i = 1, 2)$ is obtained as follows:

$$\mathsf{CN}_{P_1}(\Delta_1) = \{p, r, k, not\ y, not\ f, not\ \neg p, not\ \neg r, not\ \neg y, not\ \neg f, not\ \neg k\} = S_1 \cup \Delta_1,$$
$$\mathsf{CN}_{P_1}(\Delta_2) = \{y, f, k, not\ p, not\ r, not\ \neg p, not\ \neg r, not\ \neg y, not\ \neg f, not\ \neg k\} = S_2 \cup \Delta_2.$$

As a result, the expected result is successfully obtained since $k \in \mathsf{CN}_{P_1}(\Delta_i)$ $(i = 1, 2)$.

Example 6. Consider logic programs $P = \{a \leftarrow,\ \neg a \leftarrow,\ b \leftarrow not\ b\}$ and $Q = \{\neg a \leftarrow,\ a \leftarrow not\ b\}$ shown in [28, Example 3.4]. Both are inconsistent under the answer set semantics (resp. under the p-stable model semantics) since P has the unique answer set $S = Lit_P$ (resp. no p-stable model), while Q has no answer set (resp. only the inconsistent p-stable model $M = \{\neg a, a\}$). Correspondingly, $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{\ \ } \rangle$ has no stable assumption extension (though it has the unique complete, preferred and grounded assumption extension $\{not\ \neg b\}$ which is inconsistent since $\mathsf{CN}_P(\{not\ \neg b\}) = \{a, \neg a, not\ \neg b\}$), while $\mathbf{ABF}(P_{tr}) = \langle \mathfrak{L}, P_{tr}, \mathcal{A}_P, \overline{\ \ } \rangle$ has the unique stable assumption extension $\Delta = \{\}$ which is inconsistent since $\mathsf{CN}_{P_{tr}}(\Delta) = Lit_P = \{a, \neg a,\ b, \neg b\} = S \cup \Delta$. In contrast, $\mathbf{ABF}(Q)$ has the unique stable (resp. complete, preferred and grounded) assumption extension $\Delta' = \{not\ b,\ not\ \neg b\}$ which is inconsistent since $\mathsf{CN}_Q(\Delta') = \{a, \neg a, not\ b, not\ \neg b\} = M \cup \Delta'$, while $\mathbf{ABF}(Q_{tr})$ has no stable assumption extension.

3.3. Arguments and argument extensions in ABFs translated from EDLPs

Given an (E)DLP, it is impossible to capture its semantics by using arguments constructed based on Definition 1. The reason is as follows. For example, consider the EDLP $P = \{p \mid q \leftarrow\}$ which has two answer sets, $\{p\}$ and $\{q\}$ (or the DLP $P_D = \{p \vee q \leftarrow\}$ which has two stable models, $\{p\}$ and $\{q\}$). Then if we use Definition 1 in the ABA framework $\mathcal{F}(P)$ (or $\mathcal{F}(P_D)$) instantiated with P (or P_D), we may construct the tree which has the root labelled by $p \mid q$ (or $p \vee q$) and the unique child labelled τ together with two one-node-trees whose roots are labelled by either $not\ p$ or $not\ q$, but no other trees. As a result, since there exists no *attacks* among these three arguments, there is the unique argument extension $\mathcal{E}$ consisting of these arguments s.t. $\mathrm{Concs}(\mathcal{E}) = \{p \mid q, not\ p, not\ q\}$ (or s.t. $\mathrm{Concs}(\mathcal{E}) = \{p \vee q, not\ p, not\ q\}$) under any σ semantics in the ABF $\mathcal{F}(P)$ (or $\mathcal{F}(P_D)$). Thus two answer sets shown above can never be captured based on Definition 1. In contrast, in $\mathbf{ABF}(P)$ (or in $\mathbf{ABF}(P_D)$), we can use the inference rule [Res]. Hence, if there is the aforementioned tree whose root is labelled by $p \mid q$, thanks to the inference rule [Res], we can construct two trees furthermore, each of which has the root labelled by p (resp. q) as well as two children such that one is the child tree whose root is labelled by $p \mid q$ and the other is the child labelled by the defeasible assumption $not\ q$ (resp. $not\ p$). As a result, two arguments (i.e. trees) whose roots are labelled by either p or q attack each other, which enables us to capture two answer sets of P. The same goes for $\mathbf{ABF}(P_D)$ to capture two stable models of P_D. Thus, this example illustrates that Definition 1 has the difficulty to deal with disjunctive rules in the ABA framework $\mathcal{F}(P)$ (resp. $\mathcal{F}(P_D)$), whereas inference rules provided in $\mathbf{ABF}(P)$ (resp. $\mathbf{ABF}(P_D)$) are useful to construct arguments using disjunctive rules. (Furthermore, see details in Example 7 as shown below.)

In what follows, arguments and attacks in the ABF translated from an EDLP P are defined.

Definition 11. Let $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$ be the ABF translated from an EDLP P, where $\mathfrak{L} = \langle \mathcal{L}_{\text{EDLP}}, \Vdash \rangle$. Ψ belonging to $\mathcal{L}_{\text{EDLP}}$ is said to be a *defeasible consequence* of P and $K \subseteq \mathcal{A}_P$ if $P \cup K \Vdash \Psi$ in which any assumption contained in K is used to derive Ψ. K is said to be a *support for* Ψ *w.r.t.* P.

$P \cup K \Vdash \Psi$ addressed above is represented by a tree structure $\mathcal{T}_\Psi(K)$ as follows.

Definition 12. Let $\mathfrak{L} = \langle \mathcal{L}_{\text{EDLP}}, \Vdash \rangle$ and $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$ be the ABF translated from an EDLP P. Let $\mathcal{T}_\Psi(K)$ denote $P \cup K \Vdash \Psi$ where K is the support for a defeasible consequence Ψ w.r.t. P. In other words, $\mathcal{T}_\Psi(K)$ is a (finite) *tree with a root node labelled by* Ψ *(having a support* K*)* defined as follows.

1. The cases using no inference rules:
   (1) For $not\ \ell \in \mathcal{A}_P$, there is a one-node tree $\mathcal{T}_\Psi(K)$ whose root node is labelled by $\Psi = not\ \ell$ and $K = \{not\ \ell\}$.
   (2) For a rule $r \in P$, there is a one-node tree $\mathcal{T}_\Psi(K)$ whose root node is labelled by $\Psi = r$ and $K = \emptyset$.
2. The cases using inference rules:
   (1) **i.** For a rule $\psi \leftarrow\ \in P$, by [Ref], there is a tree $\mathcal{T}_\psi(K)$ whose root node N is labelled by ψ and N has a unique child node, namely a one-node tree $\mathcal{T}_r(\emptyset)$ where $r = \psi \leftarrow$. Then $K = \emptyset$.
   ii. For a rule $\psi \leftarrow \phi_1, \ldots, \phi_n$ in P, if for each ϕ_i $(1 \leqslant i \leqslant n)$, there exists a tree $\mathcal{T}_{\phi_i}(K_i)$ with the root node N_i labelled by ϕ_i, then by [MP], there is a tree $\mathcal{T}_\psi(K)$ with the root node N labelled by ψ and N has a child N_0 labelled by $r = \psi \leftarrow \phi_1, \ldots, \phi_n$ which is a one-node tree $\mathcal{T}_r(\emptyset)$ as well as n children N_i $(1 \leqslant i \leqslant n)$ where N_i is the root of a tree $\mathcal{T}_{\phi_i}(K_i)$. Then $K = \bigcup_i K_i$.
   (2) Let $\Phi = \psi'_1|\ldots|\psi'_m|\ell_1|\ldots|\ell_n|\psi''_1|\ldots|\psi''_k$ and $\Psi = \psi'_1|\ldots|\psi'_m|\psi''_1|\ldots|\psi''_k$, where $\ell_i \in Lit_P$ $(1 \leqslant i \leqslant n)$. If there is a tree $\mathcal{T}_\Phi(K')$ with the root node N_0 labelled by Φ, then by [Res], there is a tree $\mathcal{T}_\Psi(K)$ with the root node N labelled by Ψ and N has a child N_0 as well as n children $N_1, \ldots N_n$ each of which is a one-node tree $\mathcal{T}_{\phi_i}(\{\phi_i\})$ where $\phi_i = not\ \ell_i$ $(1 \leqslant i \leqslant n)$. Then $K = K' \cup \bigcup_{i=1}^n \{not\ \ell_i\}$.
   (3) Let $(\ell_i \ldots \psi)$[10] denote the reasoning for the case ℓ_i and $\mathcal{T}_{\ell_i}(\emptyset)$ be a one-node tree whose root is labelled by ℓ_i. Suppose that
   - there is a tree $\mathcal{T}_\Phi(K')$ whose root node N_0 is labelled by $\Phi = \ell_1|\ldots|\ell_n$; and
   - for each ℓ_i $(1 \leqslant i \leqslant n)$, there exists reasoning for a case ℓ_i such that $(\ell_i \ldots \psi)$, namely $P \cup \{\ell_i\} \cup K_i \Vdash \psi$ for $\exists K_i \subseteq \mathcal{A}_P$, which is represented by a tree $\mathcal{T}_\psi(K_i)$ constructed by newly introducing a tree $\mathcal{T}_{\ell_i}(\emptyset)$ in this definition.

   Then by [RBC], there is a tree $\mathcal{T}_\psi(K)$ with the root node N labelled by ψ and N has the child N_0 as well as n children $N_1, \ldots N_n$ where each N_i $(1 \leqslant i \leqslant n)$ is the root of a tree $\mathcal{T}_\psi(K_i)$ for the case ℓ_i. Thus $K = K' \cup \bigcup_{i=1}^n \{K_i\}$. $\square$

Given $\mathbf{ABF}(P)$, the set of all trees $\mathcal{T}_\Psi(K)$ is uniquely determined based on Definition 12.

In $\mathbf{ABF}(P)$, an *argument* is defined as a special tree $\mathcal{T}_\phi(K)$ whose root is labelled by a literal or a NAF-literal $\phi \in \mathbb{L}_P$, and the attack relation *attacks* is defined as usual.

[10]This is depicted vertically in the inference rule of [RBC].

Definition 13. Let $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$ be the ABF translated from an EDLP P and $\phi \in \mathbb{L}_P = Lit_P \cup NAF_P$. Then in $\mathbf{ABF}(P)$,

- *an argument for a conclusion (or claim) ϕ supported by* $K \subseteq \mathcal{A}_P$ ($K \vdash \phi$, for short) is a (finite) tree $\mathcal{T}_\phi(K)$ whose root node is labelled by $\phi \in \mathbb{L}_P$.
- $K_1 \vdash \phi_1$ *attacks* $K_2 \vdash \phi_2$ iff $\phi_1 = \overline{\alpha}$ for some $\alpha \in K_2$.

Notation 1. Given an EDLP P, we often use a unique name to denote an argument $K \vdash \phi$ in $\mathbf{ABF}(P)$, e.g. $a : K \vdash \phi$ is an argument with name a. With an abuse of notation [31, Notation 3], the name of an argument sometimes stands for the whole argument, for example, a denotes the argument $a : K \vdash \phi$.

Let $AF = (AR, attacks)$ be the abstract argumentation (AA) framework generated from $\mathbf{ABF}(P)$, where AR is the set of all arguments such that $a \in AR$ iff an argument $a : K \vdash \phi$ is in $\mathbf{ABF}(P)$, and $(a, b) \in attacks$ in AF iff a *attacks* b in $\mathbf{ABF}(P)$.

The semantics is also given by *argument* extensions in $\mathbf{ABF}(P)$ as follows.

Definition 14. Let $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$ be the ABF translated from an EDLP P, AR be the set of arguments generated from $\mathbf{ABF}(P)$, and $\mathcal{A}rgs \subseteq AR$. $\mathcal{A}rgs$ is *conflict-free* iff $\nexists A, B \in \mathcal{A}rgs$ such that A *attacks* B. $\mathcal{A}rgs$ *defends* an argument A iff each argument that attacks A is attacked by an argument in $\mathcal{A}rgs$. Then $\mathcal{A}rgs \subseteq AR$ is: *admissible* iff $\mathcal{A}rgs$ is conflict-free and defends all its elements; a *complete* argument extension iff $\mathcal{A}rgs$ 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 attacks every argument in $AR \setminus \mathcal{A}rgs$; an *ideal* argument extension iff it is a (subset-)maximal complete argument extension that is contained in each preferred argument extension.

Example 7. Consider the EDLP $P = \{p \mid q \leftarrow\}$. Then $\mathbf{ABF}(P)$ has four arguments A_i $(1 \leqslant i \leqslant 4)$:

$$A_1 : \{not\ q\} \vdash p, \qquad A_2 : \{not\ p\} \vdash q, \qquad A_3 : \{not\ p\} \vdash not\ p, \qquad A_4 : \{not\ q\} \vdash not\ q,$$

whose tree structures are shown in Fig. 1. Fig. 2 shows the AA framework $AF = (AR, attacks)$ for the ABF. It has two argument extensions $\mathcal{E}_1, \mathcal{E}_2$ under stable and preferred semantics as follows:

$$\mathcal{E}_1 = \{A_1, A_4\}, \quad \mathcal{E}_1 = \{A_2, A_3\}, \quad \text{where Concs}(\mathcal{E}_1) = \{p,\ not\ q\}, \quad \text{Concs}(\mathcal{E}_2) = \{q,\ not\ p\}.$$

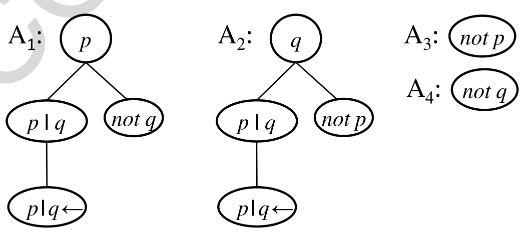

Fig. 1. Arguments of $\mathbf{ABF}(P)$ for $P = \{p \mid q \leftarrow\}$ in Ex. 7

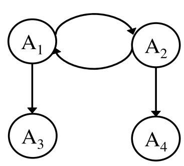

Fig. 2. $AF = (AR, attacks)$ in Ex. 7

Let $\sigma \in$ {complete, preferred, grounded, stable, ideal}. It is shown that there is a one-to-one correspondence between σ argument extensions and σ assumption extensions of $\mathbf{ABF}(P)$ for an EDLP P.

Definition 15. Let $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$ be the ABF translated from an EDLP P, and AR be the set of all arguments that can be generated from $\mathbf{ABF}(P)$. $\mathsf{Asms2Args} : \wp(\mathcal{A}_P) \to \wp(AR)$ and $\mathsf{Args2Asms} : \wp(AR) \to \wp(\mathcal{A}_P)$ are functions such that

$$\mathsf{Asms2Args}(\mathcal{A}sms) = \{K \vdash \phi \in AR \mid K \subseteq \mathcal{A}sms\},$$

$$\mathsf{Args2Asms}(\mathcal{A}rgs) = \{\alpha \in \mathcal{A}_P \mid \alpha \in K \text{ for an argument } K \vdash \phi \in \mathcal{A}rgs\}.$$

Theorem 5. *Let $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$ be the ABF translated from an EDLP P, and AR be the set of all arguments generated from $\mathbf{ABF}(P)$. The following holds.*

1. *If $\mathcal{A}sms \subseteq \mathcal{A}_P$ is a σ assumption extension, then $\mathsf{Asms2Args}(\mathcal{A}sms)$ is a σ argument extension.*
2. *If $\mathcal{A}rgs \subseteq AR$ is a σ argument extension, then $\mathsf{Args2Asms}(\mathcal{A}rgs)$ is a σ assumption extension.*

Proof. In case $\sigma =$ stable, this is proved as follows.

1. Let $\mathcal{A}sms \subseteq \mathcal{A}_P$ be a stable assumption extension and $\mathcal{A}rgs = \mathsf{Asms2Args}(\mathcal{A}sms)$.

   (i) Since $\mathcal{A}sms$ is conflict-free, every argument constructed from $\mathcal{A}sms$ (i.e. $K \vdash \phi$ where $K \subseteq \mathcal{A}sms$) does not attack any assumption in $\mathcal{A}sms$. In other words, any argument in $\mathcal{A}rgs$ does not attack any argument constructed from $\mathcal{A}sms$. This means that $\mathcal{A}rgs$ is conflict-free.

   (ii) Since $\mathcal{A}sms$ is a stable assumption extension, $\mathcal{A}sms$ attacks any assumption in $\mathcal{A}_P \setminus \mathcal{A}sms$. Let $B = (K_B \vdash \phi)$ be any argument in $AR \setminus \mathcal{A}rgs$, where $K_B \nsubseteq \mathcal{A}sms$ and $(\mathcal{A}_P \setminus \mathcal{A}sms) \cap K_B \neq \emptyset$. Then $\mathcal{A}sms$ attacks $(\mathcal{A}_P \setminus \mathcal{A}sms) \cap K_B$. Therefore $\mathcal{A}sms$ attacks B, denoting that $\mathcal{A}rgs$ attacks B. Hence $\mathcal{A}rgs$ attacks any argument in $AR \setminus \mathcal{A}rgs$.

   Due to (i) and (ii), $\mathcal{A}rgs$ is conflict-free and attacks every argument in $AR \setminus \mathcal{A}rgs$. Thus $\mathcal{A}rgs$ is a stable argument extension.
2. Let $\mathcal{A}rgs \subseteq AR$ be a stable argument extension and $\mathcal{A}sms = \mathsf{Args2Asms}(\mathcal{A}rgs)$.

   (i) $\mathcal{A}rgs$ is conflict-free since $\mathcal{A}rgs$ is a stable assumption extension. Suppose $\mathcal{A}sms$ is not conflict-free. That is, $\mathcal{A}sms$ attacks some assumption $\exists\alpha \in \mathcal{A}sms$. Then it is possible to construct some argument based on $\mathcal{A}sms$ (say $A = (K_A \vdash \phi)$ where $K_A \subseteq \mathcal{A}sms$) whose conclusion ϕ is the contrary of some assumption α in $\mathcal{A}sms$, i.e. $\phi = \overline{\alpha}$ for $\exists\alpha \in \mathcal{A}sms$. The argument A cannot be a member of $\mathcal{A}rgs$ (otherwise $\mathcal{A}rgs$ would not be conflict-free). Hence $A \in AR \setminus \mathcal{A}rgs$. Besides for $\forall\alpha \in K_A \subseteq \mathcal{A}sms$, $\mathcal{A}rgs$ does not attack α since $\mathcal{A}rgs$ is conflict-free. This means that $\mathcal{A}rgs$ cannot attack the argument $A \in AR \setminus \mathcal{A}rgs$. Hence $\mathcal{A}rgs$ is not a stable argument extension. Contradiction. Therefore $\mathcal{A}sms$ is conflict-free.

   (ii) Suppose there exists some assumption $\alpha \in \mathcal{A}_P \setminus \mathcal{A}sms$ which $\mathcal{A}sms$ does not attack. Due to $\alpha \notin \mathcal{A}sms$, there is no argument in $\mathcal{A}rgs$ whose support contains α. Hence $(\{\alpha\} \vdash \alpha) \in AR \setminus \mathcal{A}rgs$. Since $\mathcal{A}rgs$ is a stable argument extension, there exists some argument $B = (K_B \vdash \phi) \in \mathcal{A}rgs$ with $\phi = \overline{\alpha}$ and $K_B \subseteq \mathcal{A}sms$ which attacks the argument $\{\alpha\} \vdash \alpha$. This means that $\mathcal{A}rgs$ attacks $\{\alpha\} \vdash \alpha$. In other words, $\mathcal{A}sms$ attacks $\alpha \in \mathcal{A}_P \setminus \mathcal{A}sms$. Contradiction.

   Due to (i) and (ii), $\mathcal{A}sms$ is conflict-free and it attacks every assumption $\alpha \in \mathcal{A}_P \setminus \mathcal{A}sms$. Therefore $\mathcal{A}sms$ is a stable assumption extension.

The other cases are proved in a similar way to the proof in [32, Theorem 2]. $\square$

When an ELP P with no disjunction is given, an argument $K \vdash \phi$ in the ABA framework $\mathcal{F}(P)$ is the tree constructed in accordance with Definition 1, whereas in $\mathbf{ABF}(P)$, an argument $K \vdash \phi$ is the tree $\mathcal{T}_\Psi(K)$ defined by Definition 12, which is constructed by using only the inference rule [MP] (including [Ref]). Though the tree structure of $K \vdash \phi$ in the ABA framework $\mathcal{F}(P)$ is different from that of $K \vdash \phi$ in $\mathbf{ABF}(P)$, they are semantically equivalent as follows.

Proposition 7. *Let P be an ELP, $\phi \in \mathbb{L}_P$, and $K \subseteq \mathcal{A}_P = \mathrm{NAF}_P$. Then $K \vdash \phi$ is an argument of the ABA framework $\mathcal{F}(P) = \langle \mathcal{L}_P, P, \mathcal{A}_P, \overline{} \rangle$ iff there is an argument $K \vdash \phi$ of $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$, where $\mathfrak{L} = \langle \mathcal{L}_{\mathrm{EDLP}}, \Vdash \rangle$.*

Proof. We denote by $\vdash_{\mathrm{MP}}$ derivability using modus ponens on $\leftarrow$ as the only inference rule in the ABA framework $\mathcal{F}(P)$. When used on $P \cup K$ for $K \subseteq NAF_P$, $\vdash_{\mathrm{MP}}$ treats NAF literals purely syntactically [31]. On the other hand, we denote by $\Vdash_{\mathrm{MP}}$ derivability using only the inference rule [MP] in $\mathbf{ABF}(P)$. Then

$K \vdash \phi$ is an argument in the ABA framework $\mathcal{F}(P)$, where K is a support for $\phi \in \mathbb{L}_P = \mathcal{L}_P$
iff $P \cup K \vdash_{\mathrm{MP}} \phi$ w.r.t. a support $K \subseteq \mathcal{A}_P$ for ϕ (due to [31, Lemma 2])
iff $P \cup K \Vdash_{\mathrm{MP}} \phi$ w.r.t. a support $K \subseteq \mathcal{A}_P$ for ϕ in $\mathbf{ABF}(P)$
iff $P \cup K \Vdash \phi$ w.r.t. a support $K \subseteq \mathcal{A}_P$ for ϕ in $\mathbf{ABF}(P)$
iff $K \vdash \phi$ is an argument in $\mathbf{ABF}(P)$. $\square$

The following proposition denotes that, given an ELP, the same abstract argumentation framework is obtained regardless of whether arguments are constructed according to either Definition 1 or Definition 12.

Proposition 8. *Given an ELP P, the abstract argumentation (AA) framework generated from the ABA framework $\mathcal{F}(P)$ coincides with the AA framework generated from $\mathbf{ABF}(P)$.*

Proof. Given an ELP P, let $AF_1 = (AR_1, attacks_1)$ be the AA framework generated from the ABF $\mathcal{F}(P) = \langle \mathcal{L}_P, P, \mathcal{A}_P, \overline{} \rangle$ and $AF_2 = (AR_2, attacks_2)$ be the AA framework generated from $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$.

(1) Then, $a \in AR_1$ iff there exists $(a : K \vdash \phi)$ in $\mathcal{F}(P)$ where $K \subseteq \mathcal{A}_P$ and $\phi \in \mathcal{L}_P$
iff there exists $(a : K \vdash \phi)$ in $\mathbf{ABF}(P)$ where $K \subseteq \mathcal{A}_P$ and $\phi \in \mathbb{L}_P$ due to Proposition 7
iff $a \in AR_2$.
(2) $(a, b) \in attacks_1$ iff a *attacks* b in $\mathcal{F}(P)$ iff $(a : K \vdash \phi)$ *attacks* $(b : K' \vdash \phi')$ in $\mathcal{F}(P)$
iff there exist $(a : K \vdash \phi)$ and $(b : K' \vdash \phi')$ s.t. *not* $\phi \in K'$ in $\mathcal{F}(P)$
iff there exist $(a : K \vdash \phi)$ and $(b : K' \vdash \phi')$ s.t. *not* $\phi \in K'$ in $\mathbf{ABF}(P)$ due to Proposition 7
iff $(a : K \vdash \phi)$ *attacks* $(b : K' \vdash \phi')$ in $\mathbf{ABF}(P)$
iff a *attacks* b in $\mathbf{ABF}(P)$ iff $(a, b) \in attacks_2$.

Thanks to (1) and (2), $AR_1 = AR_2$ as well as $attacks_1 = attacks_2$ hold. Hence $AF_1 = AF_2$. $\square$

3.4. Correspondence between answer sets of an EDLP and stable argument extensions

In $\mathbf{ABF}(P)$ translated from an EDLP P, the *conclusion* of a set of arguments $\mathcal{E}$ is defined as:

$$\mathrm{Concs}(\mathcal{E}) = \big\{\phi \in \mathbb{L}_P \mid \phi \text{ is a conclusion (or claim) of an argument contained in } \mathcal{E}\big\}.$$

First of all, we show there is a one-to-one correspondence between answer sets of an NDP P and stable argument extensions of the ABF translated from P.

Theorem 6. *Let $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$ be the ABF translated from an NDP P where $\mathfrak{L} = \langle \mathcal{L}_{\mathrm{EDLP}}, \Vdash \rangle$. Let $M \subseteq HB_P$ and $\mathcal{E} \subseteq AR$, where AR is the set of all arguments generated from $\mathbf{ABF}(P)$. Then M is an answer set of an NDP P iff there is a stable argument extension $\mathcal{E}$ of $\mathbf{ABF}(P)$ such that $M \cup \Delta_M =$ $\mathsf{Concs}(\mathcal{E}) = \mathsf{CN}_P(\Delta_M)$, where $\Delta_M = \{not\ p \mid p \in (HB_P \setminus M)\}$ is a stable assumption extension of $\mathbf{ABF}(P)$.*

Proof. Let $NAF_P = \{not\ p \mid p \in HB_P\}$ and $\Delta_M \subseteq NAF_P$. Firstly we show that due to the form of inference rules, $P \cup \Delta_M \Vdash not\ p$ iff $not\ p \in \Delta_M$. In other words,

$$not\ p \in \mathsf{Cn}_{\mathfrak{L}}(P \cup \Delta_M) \quad \text{iff} \quad not\ p \in \Delta_M, \tag{4}$$

which is needed below to prove the equivalences given in this theorem.

$\Rightarrow$: Let M be an answer set of an NDP P. Then there exists the stable assumption extension Δ_M of $\mathbf{ABF}(P)$ such that $\Delta_M = \{not\ p \mid p \in (HB_P \backslash M)\}$ due to Proposition 3*(i)*. Hence due to Theorem 5, there exists the stable argument extension $\mathcal{E} = \mathsf{Asms2Args}(\Delta_M) = \{K \vdash \phi \mid K \subseteq \Delta_M,\ \mathbb{L}_P = HB_P \cup NAF_P\}$. Thus for M, Δ_M and $\mathcal{E} = \mathsf{Asms2Args}(\Delta_M)$, it holds that

$$\begin{aligned}&\mathsf{Concs}(\mathcal{E}) = \{\phi \in \mathbb{L}_P \mid K \vdash \phi \textit{ is constructed from } \Delta_M,\ \textit{where } K \subseteq \Delta_M \textit{ and } \phi \in \mathbb{L}_P = HB_P \cup NAF_P\}\\ &= \{\phi \in \mathbb{L}_P \mid P \cup \Delta_M \Vdash \phi\} = \{\phi \in \mathbb{L}_P \mid \phi \in \mathsf{Cn}_{\mathfrak{L}}(P \cup \Delta_M)\} = M \cup \Delta_M. \text{ (due to Lemma 2 and (4))}\end{aligned}$$

$\Leftarrow$: Let $\mathcal{E}$ be a stable argument extension of $\mathbf{ABF}(P)$ translated from an NDP P. Then due to Theorem 5, there is the stable assumption extension Δ of $\mathbf{ABF}(P)$ such that $\Delta = \mathsf{Args2Asms}(\mathcal{E}) = \{\alpha | \alpha \in K \text{ for } K \vdash \phi \text{ in } \mathcal{E}\} = \bigcup_i K_i$ for $K_i \vdash \phi_i \in \mathcal{E}$. Moreover, corresponding to this stable assumption extension Δ, there is the answer set M of the NDP P such that $M = \{p \in HB_P |\ not\ p \notin \Delta)\}$ due to Proposition 3*(ii)*, while for this answer set M, $\Delta_M = \{not\ p \in NAF_P | p \notin M\}$ is the stable assumption extension of $\mathbf{ABF}(P)$ due to Proposition 3*(i)*. Thus obviously $\Delta_M = \Delta$. Then for a stable argument extension $\mathcal{E}$, $\Delta = \bigcup_i K_i$ s.t. $K_i \vdash \phi_i \in \mathcal{E}$, $M = \{p \in HB_P |\ not\ p \notin \Delta\}$ and $\Delta_M = \Delta$, it holds that

$$\begin{aligned}\mathsf{Concs}(\mathcal{E}) &= \{\phi \mid K \vdash \phi \in \mathcal{E} \textit{ for } \phi \in \mathbb{L}_P = HB_P \cup NAF_P\}\\ &= \{\phi \in \mathbb{L}_P \mid P \cup \Delta \Vdash \phi \textit{ for } \Delta = \textstyle\bigcup_i K_i \textit{ where } K_i \vdash \phi_i \in \mathcal{E}\}\\ &= \{\phi \in \mathbb{L}_P \mid P \cup \Delta_M \Vdash \phi \textit{ for } M = \{p \in HB_P |\ not\ p \notin \Delta_M = \Delta\}\}\\ &= \{\phi \in \mathbb{L}_P \mid \phi \in \mathsf{Cn}_{\mathfrak{L}}(P \cup \Delta_M)\} = M \cup \Delta_M. \quad (\text{due to Lemma 2 and (4)})\end{aligned}$$

□

Based on Theorem 6, we show that there is a one-to-one correspondence between answer sets (resp. p-stable models) of an EDLP P and stable *argument* extensions of $\mathbf{ABF}(P_{tr})$ (resp. $\mathbf{ABF}(P)$) as follows, though Propositions 5 and 4 show the similar correspondences for stable *assumption* extensions of the respective ABFs.

Theorem 7. *Let $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$ be the ABF translated from an EDLP P. Let $M \subseteq Lit_P$ and $\mathcal{E} \subseteq AR$, where AR is the set of all arguments generated from $\mathbf{ABF}(P)$. Then M is a p-stable model of an EDLP P iff there is a stable argument extension $\mathcal{E}$ of $\mathbf{ABF}(P)$ such that $M \cup \Delta_M = \mathsf{Concs}(\mathcal{E}) =$ $\mathsf{CN}_P(\Delta_M)$, where Δ_M is a stable assumption extension of $\mathbf{ABF}(P)$.*

Proof. Let P^+ be the NDP which is the positive form [28] of an EDLP P. Like the proof of Proposition 4, for a set $S \subseteq Lit_P$, let S^+ be the set obtained by replacing each negative literal $\neg L$ in S with the newly introduced atom L'. Thus Herbrand base HB_{P^+} of the NDP P^+ is $(Lit_P)^+$. Then for $\mathbf{ABF}(P)$ translated from an EDLP P and $\mathbf{ABF}(P^+)$ translated from the NDP P^+, it holds that, there is a stable argument extension $\mathcal{E}$ of $\mathbf{ABF}(P)$ if and only if there is a stable argument extension $\mathcal{E}^+$ of $\mathbf{ABF}(P^+)$ such that

$$\mathrm{Concs}\big(\mathcal{E}^+\big) = \mathrm{Concs}(\mathcal{E})^+. \tag{5}$$

Moreover, due to Lemma 3,

$$M \text{ is a p-stable model of an EDLP } P \text{ iff } M^+ \text{ is an answer set of the NDP } P^+. \tag{6}$$

On the other hand, based on Theorem 6 as well as (5), it holds that,

$$M^+ \text{ is an answer set of the NDP } P^+$$

iff there is a stable argument extension $\mathcal{E}^+$ along with a stable assumption extension Δ_{M^+} of $\mathbf{ABF}(P^+)$

$$\begin{aligned}
&\text{such that } M^+ \cup \Delta_{M^+} = \mathrm{Concs}\big(\mathcal{E}^+\big),\\
&\quad \text{where } \Delta_{M^+} = \big\{ not\ a \mid a \in HB_{P^+} \setminus M^+ \big\} = \big\{ not\ a \mid a \in Lit_P^+ \setminus M^+ \big\} = \Delta_M^+,\\
&\quad M^+ \cup \Delta_{M^+} = (M \cup \Delta_M)^+ \text{ for } M \subseteq Lit_P, \text{ and } \Delta_{M^+} = \Delta_M^+
\end{aligned}$$

iff there is a stable argument extension $\mathcal{E}$ along with a stable assumption extension Δ_M of $\mathbf{ABF}(P)$

$$\text{such that } M \cup \Delta_M = \mathrm{Concs}(\mathcal{E}), \text{ where } \Delta_M = \{ not\ L \mid L \in Lit_P \setminus M \}. \tag{7}$$

Hence from (6) and (7), it follows that, M is a p-stable model of an EDLP P iff there is a stable argument extension $\mathcal{E}$ along with a stable assumption extension Δ_M of $\mathbf{ABF}(P)$ such that $M \cup \Delta_M = \mathrm{Concs}(\mathcal{E})$. $\square$

Theorem 8. *Let P be an EDLP and $\mathbf{ABF}(P_{tr}) = \langle \mathfrak{L}, P_{tr}, \mathcal{A}_P, \overline{} \rangle$ be the ABF translated from the EDLP $P_{tr} = P \cup \{ L \leftarrow p, \neg p \mid p \in Lit_P,\ L \in Lit_P \}$, where $Lit_{P_{tr}} = Lit_P$, $\mathcal{A}_P = NAF_P = \{ not\ \ell \mid \ell \in Lit_P \}$, and $\overline{not\ \ell} = \ell$ for every $not\ \ell \in \mathcal{A}_P$. Let $S \subseteq Lit_P$ and $\mathcal{E}_{tr} \subseteq AR$ for the set AR of all arguments generated from $\mathbf{ABF}(P_{tr})$. Then S is an answer set of an EDLP P iff there is a stable argument extension $\mathcal{E}_{tr}$ of $\mathbf{ABF}(P_{tr})$ such that $S \cup \Delta_S = \mathrm{Concs}(\mathcal{E}_{tr}) = \mathsf{CN}_{P_{tr}}(\Delta_S)$, where Δ_S is a stable assumption extension of $\mathbf{ABF}(P_{tr})$.*

Proof. The following (i) and (ii) hold according to [28, Theorem 3.5] and Theorem 7 respectively.
(i) S is an answer set of an EDLP P iff S is p-stable model of P_{tr}.
(ii) S is a p-stable model of an EDLP P_{tr} iff there is a stable argument extension $\mathcal{E}_{tr}$ of $\mathbf{ABF}(P_{tr})$ such that $S \cup \Delta_S = \mathrm{Concs}(\mathcal{E}_{tr})$, where Δ_S is a stable assumption extension of $\mathbf{ABF}(P_{tr})$.
Hence this theorem follows from both (i) and (ii). $\square$

Theorems 7 and 8 for an EDLP are the generalization of Theorems 2 and 3 for an ELP respectively.

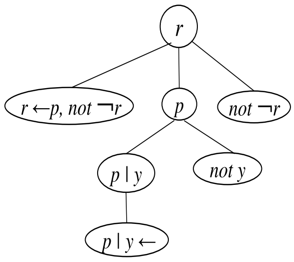

Fig. 3. A_3:$\{not\ y, not \neg r\} \vdash r$ in Example 8.

Example 8 (Cont. Example 1). To solve the *Kyoto protocol problem* in argumentation, we construct **ABF**(P_1) from $P_1 = \{r \leftarrow p, not\ \neg r,\ \ k \leftarrow r, not\ \neg k,\ \ f \leftarrow y, not\ \neg f,\ \ k \leftarrow f, not\ \neg k,\ \ p \mid y \leftarrow\}$. Arguments and *attacks* in **ABF**(P_1) are obtained as follows:

$$
\begin{aligned}
&A_1 : \{not\ y\} \vdash p, \qquad A_2 : \{not\ p\} \vdash y, \qquad A_3 : \{not\ y, not\ \neg r\} \vdash r,\\
&A_4 : \{not\ p, not\ \neg f\} \vdash f, \qquad A_5 : \{not\ y, not\ \neg r, not\ \neg k\} \vdash k,\\
&A_6 : \{not\ p, not\ \neg f, not\ \neg k\} \vdash k, \qquad A_7 : \{not\ \neg r, not\ \neg f, not\ \neg k\} \vdash k,\\
&A_8 : \{not\ p\} \vdash not\ p, \qquad A_9 : \{not\ y\} \vdash not\ y, \qquad A_{10} : \{not\ r\} \vdash not\ r,\\
&A_{11} : \{not\ f\} \vdash not\ f, \qquad A_{12} : \{not\ k\} \vdash not\ k, \qquad A_{13} : \{not\ \neg p\} \vdash not\ \neg p,\\
&A_{14} : \{not\ \neg y\} \vdash not\ \neg y, \qquad A_{15} : \{not\ \neg r\} \vdash not\ \neg r,\\
&A_{16} : \{not\ \neg f\} \vdash not\ \neg f, \qquad A_{17} : \{not\ \neg k\} \vdash not\ \neg k;\\
&attacks = \big\{(A_1, A_2), (A_1, A_4), (A_1, A_6), (A_1, A_8), (A_2, A_1), (A_2, A_3), (A_2, A_5), (A_2, A_9),\\
&\qquad\qquad (A_3, A_{10}), (A_4, A_{11}), (A_5, A_{12}), (A_6, A_{12}), (A_7, A_{12})\big\}.
\end{aligned}
$$

Fig. 3 shows A_3 which is constructed based on [Ref], [MP] and [Res]. Each A_i $(1 \leqslant i \leqslant 6)$ uses [Res]. But A_7 uses [RBC] instead of [Res]. Arguments in **ABF**(P_1) coincide with those in **ABF**$((P_1)_{tr})$.

Then **ABF**$((P_1)_{tr})$ (resp. **ABF**(P_1)) has two stable (resp. preferred) argument extensions:

$$
\begin{aligned}
\mathcal{E}_1 &= \{A_1, A_3, A_5, A_7, A_9, A_{11}\} \cup \{A_i \mid 13 \leqslant i \leqslant 17\},\\
\mathcal{E}_2 &= \{A_2, A_4, A_6, A_7, A_8, A_{10}\} \cup \{A_i \mid 13 \leqslant i \leqslant 17\},
\end{aligned}
$$

where $\mathrm{Concs}(\mathcal{E}_1) = \{p,\ r,\ k,\ not\ y,\ not\ f\} \cup U$, $\mathrm{Concs}(\mathcal{E}_2) = \{y,\ f,\ k,\ not\ p, not\ r\} \cup U$, for $U = \mathrm{Concs}\big(\{A_i \mid 13 \leqslant i \leqslant 17\}\big) = \{not\ \neg p, not\ \neg y, not\ \neg r, not\ \neg f, not\ \neg k\}$.

Hence the expected result is successfully obtained since $k \in \mathrm{Concs}(\mathcal{E}_i)$ for $\forall \mathcal{E}_i$ $(i = 1, 2)$.

On the other hand, under the grounded and ideal semantics as the skeptical semantics, **ABF**(P_1) has the unique argument extension $\mathcal{E}_3$:

$$\mathcal{E}_3 = \{A_7\} \cup \{A_i \mid 13 \leqslant i \leqslant 17\}, \quad \text{where} \quad \mathrm{Concs}(\mathcal{E}_3) = \{k\} \cup U.$$

Then as a skeptical consequence, we obtain the result such that $k \in \mathrm{Concs}(\mathcal{E}_3)$ due to $A_7 \in \mathcal{E}_3$ where the argument A_7 is constructed using [RBC], that meets our expectation based on reasoning by cases addressed in Example 1.

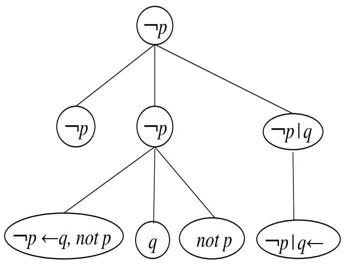

Fig. 4. A_1:$\{not\ p\} \vdash \neg p$ in Example 9.

Example 9. Consider the EDLP $P_2 = \{\neg p \mid q \leftarrow, \quad q \leftarrow \neg p, not\ \neg q, \quad \neg p \leftarrow q, not\ p\}$, where $Lit_{P_2} = \{p, q, \neg p, \neg q\}$. P_2 is not *head-cycle-free*[11] but a general EDLP [4,13]. This means that the knowledge expressed by P_2 is unlikely to be expressed by ELPs in general due to complexity results shown in [4,13]. P_2 has the unique answer set $S = \{\neg p, q\}$, which is its unique p-stable model.

In contrast, $\mathbf{ABF}(P_2)$ translated from P_2 has the unique stable assumption extension $\Delta_S = \{not\ p, not\ \neg q\}$, while it has the following arguments:

$$A_1 : \{not\ p\} \vdash \neg p, \quad A_2 : \{not\ \neg q\} \vdash q, \quad A_3 : \{not\ q\} \vdash \neg p, \quad A_4 : \{not\ \neg p\} \vdash q,$$
$$A_5 : \{not\ \neg q, not\ p\} \vdash \neg p, \quad A_6 : \{not\ p, not\ \neg q\} \vdash q, \quad A_7 : \{not\ \neg p, not\ p\} \vdash \neg p,$$
$$A_8 : \{not\ q,\ not\ \neg q\} \vdash q, \quad A_9 : \{not\ \neg p\} \vdash not\ \neg p, \quad A_{10} : \{not\ q\} \vdash not\ q,$$
$$A_{11} : \{not\ p\} \vdash not\ p, \quad A_{12} : \{not\ \neg q\} \vdash not\ \neg q, \quad \text{and}$$

$$\begin{aligned} attacks = \big\{ & (A_1, A_4), (A_1, A_7), (A_1, A_9), (A_2, A_3), (A_2, A_8), (A_2, A_{10}), (A_3, A_4), (A_3, A_7), \\ & (A_3, A_9), (A_4, A_3), (A_4, A_8), (A_4, A_{10}), (A_5, A_4), (A_5, A_7), (A_5, A_9), (A_6, A_3), \\ & (A_6, A_8), (A_6, A_{10}), (A_7, A_4), (A_7, A_7), (A_7, A_9), (A_8, A_3), (A_8, A_8), (A_8, A_{10}) \big\}. \end{aligned}$$

Fig. 4 shows the tree structure of the argument $A_1 : \{not\ p\} \vdash \neg p$ which is constructed based on the inference rules: [Ref], [MP] and [RBC]. In this case, arguments in $\mathbf{ABF}(P_2)$ coincide with those in $\mathbf{ABF}((P_2)_{tr})$. Then $\mathbf{ABF}(P_2)$ (resp. $\mathbf{ABF}((P_2)_{tr})$) has the following unique stable argument extension:

$$\mathcal{E} = \{A_1, A_2, A_5, A_6, A_{11}, A_{12}\}, \quad \text{where} \quad \mathrm{Concs}(\mathcal{E}) = \{\neg p, q, not\ p, not\ \neg q\} = S \cup \Delta_S$$

for the answer set S of P_2 and the stable assumption extension Δ_S of $\mathbf{ABF}(P_2)$.

Example 10. Consider the EDLP $P_3 = \{\neg a \mid b \leftarrow not\ \neg b, \quad a \leftarrow not\ c\}$.

It has the unique answer set $S = \{a, b\}$, while it has two p-stable models $M_1 = \{a, b\} = S$ and $M_2 = \{a, \neg a\}$, where M_1 is consistent but M_2 is inconsistent due to $M_2(a) = \top$.

Let $\mathbf{ABF}(P_3)$ be the ABF translated from P_3, which has arguments and *attacks* as follows:

$$A_1 : \{not\ c\} \vdash a, \quad A_2 : \{not\ b, not\ \neg b\} \vdash \neg a, \quad A_3 : \{not\ \neg a, not\ \neg b\} \vdash b,$$

[11] The *dependency graph* of an EDLP P is a directed graph where each literal is a node and where there is an edge from L to L' iff there is a rule in which a literal L is not preceded by the *not* operator in the body and L' appears in the head. An EDLP is *head-cycle free* iff its dependency graph does not contain directed cycles that go through two literals that belong to the head of the same rule [4].

$A_4 : \{not\ a\} \vdash not\ a,\quad A_5 : \{not\ b\} \vdash not\ b,\quad A_6 : \{not\ c\} \vdash not\ c,$
$A_7 : \{not\ \neg a\} \vdash not\ \neg a,\quad A_8 : \{not\ \neg b\} \vdash not\ \neg b,$
$A_9 : \{not\ \neg c\} \vdash not\ \neg c,$ and $attacks = \big\{(A_1, A_4), (A_2, A_3), (A_2, A_7), (A_3, A_2), (A_3, A_5)\big\}.$

Then $\mathbf{ABF}(P_3)$ has two stable argument extensions $\mathcal{E}_1, \mathcal{E}_2$ as follows.

$$\mathcal{E}_1 = \{A_1, A_3, A_6, A_7, A_8, A_9\}, \qquad \mathcal{E}_2 = \{A_1, A_2, A_5, A_6, A_8, A_9\},$$

$$\text{where}\quad \begin{aligned} \mathrm{Concs}(\mathcal{E}_1) &= \{a,\ b,\ not\ c,\ not\ \neg a,\ not\ \neg b,\ not\ \neg c\} = M_1 \cup \Delta_{M_1},\\ \mathrm{Concs}(\mathcal{E}_2) &= \{a,\ \neg a,\ not\ b,\ not\ c,\ not\ \neg b,\ not\ \neg c\} = M_2 \cup \Delta_{M_2},\end{aligned}$$

in which Δ_{M_1} and Δ_{M_2} are stable assumption extensions of $\mathbf{ABF}(P_3)$ such that

$$\Delta_{M_1} = \{not\ c, not\ \neg a, not\ \neg b, not\ \neg c\}, \qquad \Delta_{M_2} = \{not\ b, not\ c, not\ \neg b, not\ \neg c\}.$$

In contrast, $\mathbf{ABF}((P_3)_{tr})$ has the arguments A_i $(1 \leqslant i \leqslant 9)$ along with A_j $(10 \leqslant j \leqslant 15)$ shown below:

$A_{10} : \{not\ b, not\ \neg b, not\ c\} \vdash a,\quad A_{11} : \{not\ b, not\ \neg b, not\ c\} \vdash b,$
$A_{12} : \{not\ b, not\ \neg b, not\ c\} \vdash c,\quad A_{13} : \{not\ b, not\ \neg b, not\ c\} \vdash \neg a,$
$A_{14} : \{not\ b, not\ \neg b, not\ c\} \vdash \neg b,\quad A_{15} : \{not\ b, not\ \neg b, not\ c\} \vdash \neg c.$

As a result, $\mathbf{ABF}((P_3)_{tr})$ has the unique stable argument extensions as follows:

$$\mathcal{E}_{tr} = \{A_1, A_3, A_6, A_7, A_8, A_9\} = \mathcal{E}_1,$$

$$\text{where}\quad \mathrm{Concs}(\mathcal{E}_{tr}) = \{a, b, not\ c, not\ \neg a, not\ \neg b, not\ \neg c\} = S \cup \Delta_S,$$

for the unique answer set S of P_3 and the unique stable assumption extension Δ_S of $\mathbf{ABF}((P_3)_{tr})$.

Remark 2. In Example 8, [Ref] is used to construct each A_i $(1 \leqslant i \leqslant 7)$. Thus we need [Ref]. Without [MP], we cannot build A_i $(3 \leqslant i \leqslant 7)$, which means we cannot infer k. Thus we need [MP]. In Example 9, [RBC] is used to construct A_1, A_2, A_5, A_6. Hence without [RBC], we cannot construct these arguments, which means $\mathbf{ABF}(P_2)$ has no stable extension. Thus we need [RBC]. In Example 10, without [Res], we cannot construct A_3 along with A_2. Thus we need [Res].

3.5. Correspondence between answer sets of a consistent EDLP and consistent stable extensions

Rationality postulates are defined in $\mathbf{ABF}(P)$ translated from an EDLP P like Definition 2. In what follows, we show that such $\mathbf{ABF}(P)$ always satisfies the *closure-property* (or direct consistency postulate [6]) under the stable semantics.

Definition 16 (Rationality postulates). Given an EDLP P, $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$ is said to satisfy the *consistency-property* (resp. the *closure-property*) under the σ semantics if for each σ argument extension $\mathcal{E}$ of the AA framework $AF_{\mathcal{F}}$ generated from $\mathcal{F} = \mathbf{ABF}(P)$, $\mathrm{Concs}(\mathcal{E})$ is consistent (resp. $\mathrm{Concs}(\mathcal{E})$ is closed w.r.t. CN_P).

Theorem 9. *Let $\mathcal{F}$ be $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$ translated from an EDLP P and $\mathcal{E}$ be a stable argument extension of $AF_{\mathcal{F}}$ generated from $\mathcal{F} = \mathbf{ABF}(P)$.*

(1) *$\mathcal{F}$ satisfies* the closure-property *under the stable semantics.*

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

Proof.

(1) Let M be a p-stable model of P and $\mathcal{E}$ be a stable argument extension of $\mathcal{F} = \mathbf{ABF}(P)$ satisfying $M \cup \Delta_M = \text{Concs}(\mathcal{E})$. Then

$$\mathsf{CN}_P(M \cup \Delta_M) = \{\phi \in \mathbb{L}_P | P \cup M \cup \Delta_M \Vdash \phi\} = \Delta_M \cup \{\ell \in Lit_P | P \cup M \cup \Delta_M \Vdash \ell\}. \quad (8)$$

Due to Lemma 4 and the transitive closure property of $\mathsf{Cn}_{\mathfrak{L}}$, it holds that, for $\ell \in Lit_P$,

$$\ell \in M \quad \text{iff} \quad \ell \in \mathsf{Cn}_{\mathfrak{L}}(P \cup \Delta_M) \quad \text{iff} \quad \ell \in \mathsf{Cn}_{\mathfrak{L}}\big(P \cup \Delta_M \cup \textstyle\bigcup_{\ell \in M}\{\ell\}\big) = \mathsf{Cn}_{\mathfrak{L}}(P \cup \Delta_M \cup M).$$

$$\text{Hence,} \quad \ell \in M \quad \text{iff} \quad P \cup M \cup \Delta_M \Vdash \ell \quad \text{for } \ell \in Lit_P. \quad (9)$$

Then, (9) means that $M = \{\ell \in Lit_P \mid P \cup M \cup \Delta_M \Vdash \ell\}$.
As a result, (8) leads to $\mathsf{CN}_P(M \cup \Delta_M) = M \cup \Delta_M$, namely, $\mathsf{CN}_P(\text{Concs}(\mathcal{E})) = \text{Concs}(\mathcal{E})$.
Thus $\mathcal{F}$ satisfies the closure-property under the stable semantics.

(2) $\mathcal{F}$ satisfies *the consistency-property* under the stable semantics
iff for every $\mathcal{E}$, $\text{Concs}(\mathcal{E})$ is consistent (due to Definition 16)
iff for every $\mathcal{E}$, $\mathsf{CN}_P(\text{Concs}(\mathcal{E}))$ is not contradictory (due to Definition 10)
iff for every $\mathcal{E}$, $\text{Concs}(\mathcal{E})$ is not contradictory (due to Item (1), $\mathsf{CN}_P(\text{Concs}(\mathcal{E})) = \text{Concs}(\mathcal{E})$)
iff for every $\mathcal{E}$, $\text{Concs}(\mathcal{E})$ is not contradictory w.r.t. explicit negation $\neg$
(since every stable argument extension is not contradictory w.r.t. $\overline{}$). $\square$

Given an EDLP P, the notions of consistent argument extensions and consistency in $\mathbf{ABF}(P)$ are defined like [33, Definitions 6, 7] as follows.

Definition 17 (Consistent argument extensions). Given an EDLP P, let $\mathcal{E}$ be a σ argument extension of $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$. Then $\mathcal{E}$ is said to be *consistent* if $\text{Concs}(\mathcal{E})$ is not contradictory w.r.t. $\neg$; otherwise it is *inconsistent*.

Definition 18 (Consistency in ABFs translated from EDLPs). Given an EDLP P, $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$ is said to be *consistent* under σ semantics if $\mathbf{ABF}(P)$ has a consistent σ argument extension (or a consistent σ assumption extension); otherwise it is *inconsistent*.

We show that there is a one-to-one correspondence between answer sets of a consistent EDLP P and the consistent stable argument extensions of $\mathbf{ABF}(P)$ translated from P, which is a generalization of Theorem 4 for a consistent ELP. To prove it, we provide the following lemma regarding a consistent EDLP.

Lemma 5. *Let P be an EDLP. M is a* consistent *answer set of P iff there is a* consistent *p-stable model M of P.*

Proof. $\Leftarrow$: Let S be a consistent p-stable model of P. Then S does not contain a pair of complementary literals. Since S is also a p-stable model of the *reduct* P^S according to Definition 5, S is a minimal set satisfying the condition (i) for P^S which is the not-free EDLP. Then since S does not contain a pair of complementary literals, S is also a minimal set satisfying both conditions (i) and (ii) for P^S. This denotes that S is the answer set of P^S which does not contain a pair of complementary literals. Thus S is the answer set of P^S and it is not Lit_P. Hence since the answer set S of P^S which is not Lit_P is the answer set of P, S is the consistent answer set of P.

$\Rightarrow$: The converse is proved in a similar way. □

Hereby given a consistent EDLP, we can obtain the following proposition and theorem.

Proposition 9. *Let P be a* consistent *EDLP and $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$ be the ABF translated from P. Then S is an answer set of P iff there is a* consistent *stable assumption extension $\Delta_S = \{not\ \ell \mid \ell \in (Lit_P \setminus S)\}$ of $\mathbf{ABF}(P)$.*

Proof. Suppose that S is an answer set of a consistent EDLP P. Then based on Definition 5, S is a consistent answer set of P, which means that S is a consistent p-stable model of P due to Lemma 5. Then corresponding to a p-stable model S of P which is consistent, there is a stable assumption extension $\Delta_S = \{not\ \ell \mid \ell \in (Lit_P \setminus S)\}$ of $\mathbf{ABF}(P)$ where $\mathsf{CN}_P(\Delta_S) = S \cup \Delta_S$ based on Proposition 4 and Proposition 6. Now, since S is a consistent answer set, $\mathsf{CN}_P(\Delta_S) = S \cup \Delta_S$ is not contradictory according to Definition 10. This means that Δ_S is consistent. Therefore, there is a consistent stable assumption extension $\Delta_S = \{not\ \ell \mid \ell \in (Lit_P \setminus S)\}$ of $\mathbf{ABF}(P)$. The converse is also proved in a similar way. □

Theorem 10. *Let P be a* consistent *EDLP and $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$ be the ABF translated from P. Then S is an answer set of P iff there is a* consistent *stable argument extension $\mathcal{E}$ of $\mathbf{ABF}(P)$ such that $S \cup \Delta_S = \mathrm{Concs}(\mathcal{E}) = \mathsf{CN}_P(\Delta_S)$, where Δ_S is the consistent stable assumption extension of $\mathbf{ABF}(P)$.*

Proof. Suppose that $\mathcal{E}$ is a consistent stable argument extension of $\mathbf{ABF}(P)$. Then $\mathrm{Concs}(\mathcal{E})$ is consistent, i.e. not contradictory w.r.t. $\neg$ due to Definition 10 since $\mathcal{E}$ is conflict-free. Moreover, based on Theorem 7, there is the p-stable model S of P corresponding to this $\mathcal{E}$ such that $S \cup \Delta_S = \mathrm{Concs}(\mathcal{E})$, where Δ_S is a stable assumption extension of $\mathbf{ABF}(P)$. Since $\mathrm{Concs}(\mathcal{E})$ is not contradictory w.r.t. $\neg$, in other words, it does not contain a pair of complementary literals, $\mathrm{Concs}(\mathcal{E}) = S \cup \Delta_S$ as well as the p-stable model S are consistent. Hence due to Lemma 5, S is the consistent answer set of P. As a result, S is the answer set of the consistent EDLP P. The converse is also proved in a similar way. □

Corollary 2. *Let P be a* consistent *EDLP. The following holds.*

(1) $\mathcal{E}$ is a consistent stable extension of $\mathbf{ABF}(P)$ iff $\mathcal{E}$ is a stable extension of $\mathbf{ABF}(P_{tr})$.

(2) $\mathbf{ABF}(P_{tr})$ satisfies the rationality postulates under the stable semantics.

Proof. (1) follows from Theorem 10 as well as Theorem 8 for a consistent EDLP P. (2) directly follows from (1). □

Example 11 (Innocent unless proved guilty). Consider the EDLP P_4 [26], which states that everyone is pronounced not guilty unless proven otherwise:

$$P_4 = \{innocent \mid guilty \leftarrow charged, \quad \neg guilty \leftarrow not\ proven, \quad charged \leftarrow\}.$$

Let i, g, c, p be the abbreviations for *innocent, guilty, charged, proven* respectively. P_4 has the unique answer set $S = \{c, i, \neg g\}$, where p (i.e. *proven*) is interpreted "*unknown*" under the answer set semantics. In contrast, P_4 has two p-stable models, $M_1 = \{c, i, \neg g\}$ and $M_2 = \{c, g, \neg g\}$. $M_1 = S$ is consistent whose truth values are $M_1(c) = \mathbf{t}$, $M_1(i) = \mathbf{t}$, $M_1(g) = \mathbf{f}$, $M_1(p) = \bot$, while M_2 is inconsistent due to $M_2(g) = \top$. To solve this problem in argumentation, we construct $\mathbf{ABF}(P_4)$, which has arguments:

$A_1 : \{\} \vdash c$, $A_2 : \{not\ g\} \vdash i$, $A_3 : \{not\ i\} \vdash g$, $A_4 : \{not\ p\} \vdash \neg g$,
$A_5 : \{not\ i\} \vdash not\ i$, $A_6 : \{not\ g\} \vdash not\ g$, $A_7 : \{not\ c\} \vdash not\ c$,
$A_8 : \{not\ p\} \vdash not\ p$, $A_9 : \{not\ \neg i\} \vdash not\ \neg i$, $A_{10} : \{not\ \neg g\} \vdash not\ \neg g$,
$A_{11} : \{not\ \neg c\} \vdash not\ \neg c$, $A_{12} : \{not\ \neg p\} \vdash not\ \neg p$,
and $attacks = \{(A_1, A_7), (A_2, A_3), (A_2, A_5), (A_3, A_2), (A_3, A_6), (A_4, A_{10})\}$, where $|attacks| = 6$.

Then $\mathbf{ABF}(P_4)$ has two stable argument extensions, $\mathcal{E}_1$ and $\mathcal{E}_2$:

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

where $\mathrm{Concs}(\mathcal{E}_1) = \{c, i, \neg g, not\ g, not\ p, not\ \neg i, not\ \neg c, not\ \neg p\}$ with $\mathrm{Concs}(\mathcal{E}_1) \cap Lit_{P_4} = S$,
$\mathrm{Concs}(\mathcal{E}_2) = \{c, g, \neg g, not\ i, not\ p, not\ \neg i, not\ \neg c, not\ \neg p\}$ with $\mathrm{Concs}(\mathcal{E}_2) \cap Lit_{P_4} = M_2$.

Thus $\mathcal{E}_1$ is consistent but $\mathcal{E}_2$ is inconsistent. Hence $\mathbf{ABF}(P_4)$ is consistent under the stable semantics.

In contrast, $\mathbf{ABF}((P_4)_{tr})$ has the unique stable argument extension $\mathcal{E}_1$ due to six additionally introduced arguments to $A_i (1 \leqslant i \leqslant 12)$ and $|attacks| = 26$.

Using $\mathbf{ABF}(P_4)$, we can decide that the attorney-at-law having the argument A_4 for the claim $\neg g$ wins and the prosecutor having A_3 for g loses since $A_4 \in \mathcal{E}_1$ and $A_3 \notin \mathcal{E}_1$ for its unique consistent extension $\mathcal{E}_1$. Therefore $\neg g$ is decided.

4. Relation to nonmonotonic reasoning

4.1. Correspondence between disjunctive default logic and assumption-based argumentation

A disjunctive default theory (*ddt*, for short) [16] is a set of disjunctive defaults of the form:

$$\frac{\alpha : \ \beta_1, \dots, \beta_m}{\gamma_1 | \dots | \gamma_n},$$

where $\alpha, \beta_1, \dots, \beta_m, \gamma_1, \dots, \gamma_n (m, n \geqslant 0)$ are quantifier-free formulas. Formula α is the prerequisite of the default, $\beta_1, \dots, \beta_m$ are its justifications, and $\gamma_1, \dots, \gamma_n$ are its consequents. If the prerequisite α in the form is the formula *true*, it will be dropped; if, in addition, $m = 0$, then we write the default as $\gamma_1 | \dots | \gamma_n$.

The semantics of a *ddt* is given by extensions defined as follows.

Definition 19 ([16, Definition 5.1])**.** Let D be a disjunctive default theory, and let E be a set of sentences. E is an extension for D if it is one of the minimal deductively closed sets of sentences E' satisfying the condition: For any ground instance having the above form of any default from D, if $\alpha \in E'$ and $\neg\beta_1, \dots, \neg\beta_m \notin E$ then, for some i (($1 \leqslant i \leqslant n$), then, $\gamma_i \in E'$. A theorem is a sentence that belongs to all extensions.

Observe that for standard (nondisjunctive) default theories, this definition gives Reiter's extensions [27].

The definition of an extension for a *ddt* are also described based on the concept of *reduct* [16]. To this end, a *disjunctive rule* of the form $\frac{\alpha}{\gamma_1|\ldots|\gamma_n}$ is defined. Then it is said that a theory E is *closed under a disjunctive rule* if, whenever $\alpha \in E$, then there exist i, $1 \leqslant i \leqslant n$, $\gamma_i \in E$.

Definition 20 ([16, Definition 5.2])**.** Let D be a *ddt* and let E be a set of sentences. The *reduct* of D w.r.t. E, denoted D^E, is the set of inference rules defined as follows: An inference rule $\frac{\alpha}{\gamma_1|\ldots|\gamma_n}$ is in D^E if for some $\beta_1, \ldots, \beta_m$ such that $\neg\beta_i \notin E$, $1 \leqslant i \leqslant m$, the default $\frac{\alpha:\, \beta_1,\ldots,\beta_m}{\gamma_1|\ldots|\gamma_n}$ is in D.

Using the *reduct* D^E, another definition of an extension for a *ddt* is given by the following theorem.

Theorem 11 ([16, Theorem 5.3])**.** *A set of sentences E is an extension for a ddt D if and only if E is a minimal set closed under propositional consequence and under the rules from D^E.*

In what follows, we show the semantic relationship between *ddt*s and assumption-based frameworks.

Theorem 12. *Let P be an EDLP and $\mathbf{ABF}(P_{tr})$ be the ABF translated from P_{tr}.*
S is the set of all literals from an extension of a disjunctive default theory $emb(P)$
iff there is a stable argument extension $\mathcal{E}_{tr}$ of $\mathbf{ABF}(P_{tr})$ such that $S = \mathrm{Concs}(\mathcal{E}_{tr}) \cap Lit_P$
iff there is a stable assumption extension Δ of $\mathbf{ABF}(P_{tr})$ such that $S = \mathsf{CN}_{P_{tr}}(\Delta) \cap Lit_P$.

Proof. Based on Theorem 1 [16, Theorem 7.2], this theorem directly follows from Theorem 8 for an argument extension $\mathcal{E}_{tr}$ (resp. from Proposition 5 for an assumption extension Δ). $\square$

For a consistent EDLP, the following theorem holds.

Theorem 13. *Let P be a consistent EDLP and $\mathbf{ABF}(P)$ be the ABF translated from P.*
S is the set of all literals from an extension of a disjunctive default theory $emb(P)$
iff there is a consistent stable argument extension $\mathcal{E}$ of $\mathbf{ABF}(P)$ such that $S = \mathrm{Concs}(\mathcal{E}) \cap Lit_P$
iff there is a consistent stable assumption extension Δ of $\mathbf{ABF}(P)$ such that $S = \mathsf{CN}_P(\Delta) \cap Lit_P$.

Proof. Based on Theorem 1 [16, Theorem 7.2], this theorem directly follows from Theorem 10. $\square$

Given a nondisjunctive EDLP, i.e. an ELP P (resp. a consistent ELP P), Theorem 12 (resp. Theorem 13) also holds for the disjunctive default theory $emb(P)$ and $\mathbf{ABF}(P_{tr})$ (resp. $\mathbf{ABF}(P)$). For such a special case, however, we can show the relationship between a standard (nondisjunctive) default theory which gives *Reiter's extensions* and a standard ABA framework translated from an ELP as follows.

Theorem 14. *Let P be an ELP and $\mathcal{F}(P_{tr})$ be the ABA framework (ABF) translated from P_{tr}.*
S is the set of all literals from an extension of a default theory $emb(P)$
iff there is a stable argument extension $\mathcal{E}_{tr}$ of the ABF $\mathcal{F}(P_{tr})$ such that $S = \mathrm{Concs}(\mathcal{E}_{tr}) \cap Lit_P$
iff there is a stable assumption extension Δ of the ABF $\mathcal{F}(P_{tr})$ such that $S = CN_{P_{tr}}(\Delta) \cap Lit_P$,
where $CN_{P_{tr}}$ is the consequence operator for the ABF $\mathcal{F}(P_{tr})$ defined in Definition 2.

Proof. Based on Theorem 1 [16, Theorem 7.2], this theorem directly follows from Theorem 3 for an argument extension $\mathcal{E}_{tr}$ (resp. from Proposition 13 for an assumption extension Δ). $\square$

Theorem 15. *Let P be a consistent ELP and $\mathcal{F}(P)$ be the ABA framework translated from P.*
S is the set of all literals from an extension of a default theory $emb(P)$
iff there is a consistent stable argument extension $\mathcal{E}$ of the ABF $\mathcal{F}(P)$ such that $S = \mathrm{Concs}(\mathcal{E}) \cap Lit_P$
iff there is a consistent stable assumption extension Δ of the ABF $\mathcal{F}(P)$ such that $S = CN_P(\Delta) \cap Lit_P$
where CN_P is the consequence operator for the ABF $\mathcal{F}(P)$ defined in Definition 2.

Proof. Based on Theorem 1 [16, Theorem 7.2], this theorem directly follows from Theorem 4 for an argument extension $\mathcal{E}$ (resp. from Proposition 14 for an assumption extension Δ). □

The following example shows that we can successfully obtain the expected result k of the Kyoto protocol problem based on our assumption-based framework corresponding to the *ddt* D_2 thanks to Theorem 12. It should be noted here that this solution of the problem can never be obtained based on the existing works of ABA [5,19] corresponding to the default theory D_1 as discussed in the introduction.

Example 12 (Cont. Example 1). The disjunctive default theory D_2 has two extensions E_1 and E_2 such that $k \in E_i$ $(i = 1, 2)$, where $S_{E_1} = \{p, r, k\} \subseteq E_1$ (resp. $S_{E_2} = \{y, f, k\} \subseteq E_2$) is the set of all literals from E_1 (resp. E_2). On the other hand, D_2 is the *ddt* $emb(P_1)$ in which the EDLP P_1 can be embedded. Then according to Example 8, $\mathbf{ABF}((P_1)_{tr})$ coincides with $\mathbf{ABF}(P_1)$, and it hods that $k \in \mathrm{Concs}(\mathcal{E}_i)$ $(i = 1, 2)$ for two stable argument extensions $\mathcal{E}_1$ and $\mathcal{E}_2$ of $\mathbf{ABF}(P_1)$, where $\mathrm{Concs}(\mathcal{E}_1) \cap Lit_{P_1} = S_{E_1} = \{p, r, k\}$ and $\mathrm{Concs}(\mathcal{E}_2) \cap Lit_{P_1} = S_{E_2} = \{y, f, k\}$. Similarly, according to Example 5, it hods that $k \in \mathsf{CN}_{P_1}(\Delta_i)$ $(i = 1, 2)$ for each stable assumption extension Δ_i of $\mathbf{ABF}(P_1)$, where $\mathsf{CN}_{P_1}(\Delta_i) \cap Lit_{P_1} = S_{E_i}$.

4.2. Correspondence between prioritized circumscription and assumption-based argumentation

Circumscription [20,22,23] is a form of nonmonotonic reasoning, which was proposed to formalize the human commonsense reasoning under incomplete information. Commonsense knowledge including preferences is also often used in human argumentation. Then Bondarenko et al. showed in [5, Theorem 6.7] that Herbrand models of parallel circumscription can be captured by sets of assumptions of a corresponding assumption-based framework. Nonetheless, though preferences can be handled not in parallel circumscription but in prioritized circumscription, no study has shown a correspondence between the semantics of prioritized circumscription and the ABA semantics to the best of our knowledge. In what follows, we show new results about the relationships between them.

We first review the framework of circumscription. The following definition is due to [20]. Given a first order theory T, let P and Z be joint tuples of predicate constants from T, where P is a tuple of predicate constants $P_1, \ldots, P_m$. Let $T(P, Z)$ be a theory containing P and Z. The *circumscription of P in $T(P, Z)$ with variable Z* is defined by a second order formula as follows:

$$Circum(T; P; Z) \stackrel{\text{def}}{=} T(P, Z) \wedge \neg\exists pz\big(T(p, z) \wedge p < P\big),$$

where p, z are tuples of predicate variables similar to P, Z, and $p < P$ denotes the following formula:

$$\bigwedge_{i=1}^{m} \forall x\big(p_i(x) \supset P_i(x)\big) \wedge \neg \bigwedge_{i=1}^{m} \forall x\big(P_i(x) \supset p_i(x)\big),$$

where $P = \langle P_1, \ldots, P_m\rangle$. The formula of *circumscription* expresses that P has a minimal possible extension under the condition $T(P, Z)$ where Z is allowed to vary in the process of minimization [20]. Due to the respective roles in the process of minimization, each P_i $(1 \leqslant i \leqslant m)$ is called a *minimized predicate*, while each predicate in Z is called a *variable predicate*. Q denotes the rest of the predicates occurring in T, called the *fixed predicates*. This version of circumscription is called *parallel circumscription*.

If P is decomposed into disjoint parts $P^1, \ldots, P^k$, and the members of P^i are assigned a higher priority than the members of P^j for $i < j$, then *prioritized circumscription of* $P^1 > \cdots > P^k$ *in* T *with variable* Z is denoted by $Circum(T; P^1 > \cdots > P^k; Z)$, which is also defined by a second order formula [20,23]. Parallel circumscription coincides with prioritized circumscription for $k = 1$.

The semantics of circumscription is given based on the preorder $\leqslant^{P^1>\cdots>P^k;Z}$ defined as follows.

For a structure M, let $|M|$ be its universe and $M[\![C]\!]$ the interpretation of a predicate constant C. For a tuple P of predicate constants, $M_1[\![P]\!] \subseteq M_2[\![P]\!]$ denotes $M_1[\![P_i]\!] \subseteq M_2[\![P_i]\!]$ for every P_i in P.

Definition 21 ([20]). Let $P^1, \ldots, P^k$ be k disjoint parts of minimized predicates P. For any two structures M_1, M_2, we write $M_1 \leqslant^{P^1>\cdots>P^k;Z} M_2$ if

(i) $|M_1| = |M_2|$,
(ii) $M_1[\![K]\!] = M_2[\![K]\!]$, for every constant K in Q,
(iii) a. $M_1[\![P^1]\!] \subseteq M_2[\![P^1]\!]$,
  b. For every $i \leqslant k$, if for every $1 \leqslant j \leqslant i-1$, $M_1[\![P^j]\!] = M_2[\![P^j]\!]$, then $M_1[\![P^i]\!] \subseteq M_2[\![P^i]\!]$.

$\leqslant^{P;Z}$ stands for the preorder $\leqslant^{P^1>\cdots>P^k;Z}$ for $k = 1$. Then a structure M is a model of $Circum(T; P; Z)$ *iff* M is minimal in the class of models of T with respect to $\leqslant^{P;Z}$. A structure M is a model of $Circum(T; P^1 > \cdots > P^k; Z)$ *iff* M is minimal in the class of models of T with respect to $\leqslant^{P^1>\cdots>P^k;Z}$.

In a nutshell, the idea of the circumscriptive theory is that human nonmonotonic reasoning under incomplete knowledge (e.g. commonsense knowledge) with preferences is based on the most preferable models which are minimal ones w.r.t. $\leqslant^{P^1>\cdots>P^k;Z}$ among models of the knowledge base T.

In this paper, we consider a first order theory T without function symbols. We assume that T is given by a set of clauses of the form:

$$A_1 \vee \cdots \vee A_\ell \vee \neg B_1 \vee \cdots \vee \neg B_m,$$

where A_i $(1 \leqslant i \leqslant \ell;\ \ell \geqslant 0)$ and B_j $(1 \leqslant j \leqslant m;\ m \geqslant 0)$ are atoms and every variable in the formula is assumed to be universally quantified at the front. We also restrict our attention to *Herbrand models* of T, which has the effect of introducing both the *domain closure assumption* and the *unique name assumption* into T, and then reasoning with T reduces to the propositional level. Let Σ be a set of clauses. Then $Th(\Sigma)$ stands for a set of clauses which are theorems of Σ. A clause in $Th(\Sigma)$ which is not properly subsumed[12] by any theorem in $Th(\Sigma)$ is called a *characteristic clause*. $\mu Th(\Sigma)$ denotes the set of all characteristic clauses in $Th(\Sigma)$. t stands for a tuple of ground terms occurring in the *Herbrand universe* of T. Parallel circumscription is transformed into an EDLP [29] as follows.

[12]It is said that a clause C_1 *subsumes* a clauseC_2 if $C_1\theta \subseteq C_2$ holds for some substitution θ [29].

Definition 22 ([29]). Given $Circum(T; P; Z)$, an EDLP Π_α[13] is constructed as follows, where $p_1, \dots, p_s, z_1, \dots, z_t$, and $q_1, \dots, q_u$ are atoms whose predicates belong to P, Z, and Q respectively.

(1) For any clause in T of the form:

$$p_1 \vee \dots \vee p_\ell \vee z_1 \vee \dots \vee z_m \vee q_1 \vee \dots \vee q_n \vee \neg p_{\ell+1} \vee \dots \vee \neg p_s \vee \neg z_{m+1} \vee \dots \\ \vee \neg z_t \vee \neg q_{n+1} \vee \dots \vee \neg q_u,$$

Π_α has the rule:

$$z_1 | \dots | z_m | q_1 | \dots | q_n \leftarrow p_{\ell+1}, \dots, p_s, z_{m+1}, \dots, z_t, q_{n+1}, \dots, q_u, not\ p_1, \dots, not\ p_\ell.$$

(2) For every clause in $\mu Th(T)$ of the form:

$$p_1 \vee \dots \vee p_\ell \vee q_1 \vee \dots \vee q_n \vee \neg q_{n+1} \vee \dots \vee \neg q_u,$$

Π_α has the rule: $\quad p_1 | \dots | p_\ell | q_1 | \dots | q_n \leftarrow q_{n+1}, \dots, q_u.$

(3) For any atom p, z, and q from P, Z, and Q respectively,

Π_α has the rule:

$$\neg p \leftarrow not\ p,$$
$$z \mid \neg z \leftarrow, \qquad q \mid \neg q \leftarrow .$$

The following theorem presents that there is a one-to-one correspondence between models of parallel circumscription and answer sets of Π_α.

Theorem 16 ([29]). *Let Π_α be the EDLP translated from $Circum(T; P; Z)$. Then M is a model of $Circum(T; P; Z)$ iff M is an answer set of Π_α.*

Proposition 10. *Any answer set of an EDLP Π_α is consistent.*

Proof. In Π_α, classical negation $\neg$ appears only in rules constructed according to item 3 in Definition 22. Thus any answer set of Π_α never contains both a and $\neg a$ for any atom a from P, Z and Q. $\square$

In [20, Theorem 2], the following equivalence is shown:

$$Circum\big(T; P^1 > \dots > P^k; Z\big) = \bigwedge_{i=1}^{k} Circum\big(T; P^i; P^{i+1}, \dots, P^k, Z\big). \tag{10}$$

Based on (10), prioritized circumscription is transformed into an EDLP [35] as follows.

[13] In this study, any rule from an EDLP has the form (2). So if a rule r whose head is empty, i.e. "$\leftarrow body(r)$" is generated in the transformation, then Π_α has the rule "$\alpha \leftarrow body(r), not\ \alpha$" instead of r both of which are semantically equivalent each other under the answer set semantics, where α is a newly introduced atom that does not appear in the language.

Definition 23 ([35, Definition 3.11])**.** Given prioritized circumscription:

$$Circum\big(T\big(P^1 \dots P^k, Z, Q\big); P^1 > P^2 > \dots > P^k; Z\big),$$

where $P^r (1 \leqslant r \leqslant k)$, Z, Q are tuples of predicate symbols such as $\langle (P^r)_1, \dots, (P^r)_{\ell_r}\rangle$, $\langle Z_1, \dots, Z_m\rangle$, $\langle Q_1, \dots, Q_n\rangle$, an EDLP Π_β is constructed in two steps as follows:

(1) According to (10), a given prioritized circumscription is represented by the conjunction of k parallel circumscriptions. So, let every ith $(1 \leqslant i \leqslant k)$ parallel circumscription:

$$Circum\big(T\big(P^1 \dots P^k, Z, Q\big); P^i; P^{i+1}, \dots, P^k, Z\big)$$

be transformed in such a way that all predicate symbols occurring in it are renamed by using $Pi^1, \dots, Pi^k, Zi, Qi$ instead of $P^1, \dots, P^k, Z, Q$, which leads to the ith renamed parallel circumscription as follows:

$$Circum\big(Ti; Pi^i; Pi^{i+1}, \dots, Pi^k, Zi\big), \tag{11}$$

where Ti denotes $T(Pi^1, \dots, Pi^k, Zi, Qi)$, and Pi^r, Zi, Qi are tuples of predicate symbols such as $\langle (Pi^r)_1, \dots, (Pi^r)_{\ell_r}\rangle$, $\langle (Zi)_1, \dots, (Zi)_m\rangle$, $\langle (Qi)_1, \dots, (Qi)_n\rangle$.

(2) Π_β consists of all rules from $(\Pi_\alpha)^1, \dots, (\Pi_\alpha)^k$ and Π_γ where

(a) each $(\Pi_\alpha)^i (1 \leqslant i \leqslant k)$ is an EDLP which is constructed from the ith renamed parallel circumscription (11) according to Definition 22.

(b) Π_γ is a set of rules (i.e. an ELP) as follows:
For any predicate u from P, Z, Q and any t,

$$u(t) \leftarrow u_i(t), \qquad \neg u(t) \leftarrow \neg u_i(t). \quad (1 \leqslant i \leqslant k)$$

where u and u_i stand for any predicate symbol of $(P^r)_f$, Z_g, Q_h and any one of $(Pi^r)_f$, $(Zi)_g$, $(Qi)_h$ $(1 \leqslant f \leqslant \ell_r, 1 \leqslant g \leqslant m, 1 \leqslant h \leqslant n)$.

In what follows, $\mathcal{HB}$ denotes *Herbrand base* of T and $\mathcal{HB}_e = \mathcal{HB} \cup \{\neg p \mid p \in \mathcal{HB}\}$. The following theorem shows that there is a one-to-one correspondence between models of prioritized circumscription and answer sets of Π_β.

Theorem 17. *Let Π_β be the EDLP translated from $Circum(T; P^1 > P^2 > \dots > P^k; Z)$. Then M is a model of $Circum(T; P^1 > P^2 > \dots > P^k; Z)$ iff there is a consistent answer set S of Π_β such that $M = S \cap \mathcal{HB}_e$.*

Proof. Let $AS(\Pi)$ be a set of all answer sets of a finite ground EDLP Π. We use the following notations.

- $Acircum^i \stackrel{\text{def}}{=} \bigwedge_{j=1}^{i} Circum_j$, where $Circum_i \stackrel{\text{def}}{=} Circum(T; P^i; P^{i+1}, \dots, P^k; Z)$.
- $(\Pi_\gamma)^i \stackrel{\text{def}}{=} \{\ell \leftarrow \ell^i \mid \ell \leftarrow \ell^i \in \Pi_\gamma\}$, where a literal ℓ in $Circum_i$ is renamed to the literal ℓ^i in the ith renamed $Circum(Ti; Pi^i; Pi^{i+1}, \dots, Pi^k, Zi)$.
- $(\Pi_{\alpha\gamma})^i \stackrel{\text{def}}{=} (\Pi_\alpha)^i \cup (\Pi_\gamma)^i$.

- $(\Pi_\beta)^i \stackrel{\text{def}}{=} \bigcup_{j=1}^{i}((\Pi_{\alpha\gamma})^j)$.
- $S^i_{\alpha\gamma}$ denotes an answer set of $(\Pi_{\alpha\gamma})^i$, namely $S^i_{\alpha\gamma} \in AS((\Pi_{\alpha\gamma})^i)$.
- For a set S of literals, we say that S is consistent if S does not contain both p and $\neg p$ for an atom p.

Then, M^i is a model of $Circum_i = Circum(T; P^i; P^{i+1}, \ldots, P^k; Z)$ $(1 \leqslant i \leqslant k)$
iff there is a consistent answer set M^i of $(\Pi_\alpha)^i$ translated from $Circum_i$ (due to Theorem 16)
iff there is a consistent answer set $S^i_{\alpha\gamma}$ of $(\Pi_{\alpha\gamma})^i$ such that $M^i = S^i_{\alpha\gamma} \cap \mathcal{HB}_e$. (12)

Hence, M is a model of $Circum(T; P^1 > P^2 > \ldots > P^k; Z)$
iff M is a model of $Acircum^k = \bigwedge_{j=1}^{k} Circum_j$ (due to (10))
iff there is a consistent set $S = \bigcup_{j=1}^{k} S^j_{\alpha\gamma}$ for $S^j_{\alpha\gamma} \in AS((\Pi_{\alpha\gamma})^j)$ $(1 \leqslant j \leqslant k)$ such that $M = S \cap \mathcal{HB}_e$ (due to (12))
iff there is a consistent answer set $S = \bigcup_{j=1}^{k} S^j_{\alpha\gamma}$ of $(\Pi_\beta)^k$ such that $M = S \cap \mathcal{HB}_e$
iff there is a consistent answer set S of $\Pi_\beta = (\Pi_\beta)^k$ translated from $Circum(T; P^1 > P^2 > \ldots > P^k; Z)$ such that $M = S \cap \mathcal{HB}_e$. □

Therefore thanks to Theorem 10, the semantics of prioritized circumscription (resp. parallel circumscription) can be captured by argumentation based on Theorem 17 (resp. Theorem 16) as follows.

Theorem 18. *Let Π_α be the EDLP translated from $Circum(T; P; Z)$.*
Then M is a model of $Circum(T; P; Z)$ iff there is a consistent stable argument extension $\mathcal{E}$ of $\boldsymbol{ABF}(\Pi_\alpha) = \langle \mathfrak{L}, \Pi_\alpha, \mathcal{A}_{\Pi_\alpha}, \urcorner \rangle$ *such that* $M = \mathrm{Concs}(\mathcal{E}) \cap \mathcal{HB}_e$.

Proof. Based on Theorem 16, Theorem 10, and Proposition 10,
M is a model of $Circum(T; P; Z)$ iff M is a (consistent) answer set of Π_α
iff there is a consistent stable argument extension $\mathcal{E}$ of $\mathbf{ABF}(\Pi_\alpha)$ such that $M = \mathrm{Concs}(\mathcal{E}) \cap \mathcal{HB}_e$. □

Theorem 19. *Let Π_β be the EDLP translated from $Circum(T; P^1 > P^2 > \ldots > P^k; Z)$.*
Then M is a model of $Circum(T; P^1 > P^2 > \ldots > P^k; Z)$ iff there is a consistent stable argument extension $\mathcal{E}$ of $\boldsymbol{ABF}(\Pi_\beta) = \langle \mathfrak{L}, \Pi_\beta, \mathcal{A}_{\Pi_\beta}, \urcorner \rangle$ *such that* $M = \mathrm{Concs}(\mathcal{E}) \cap \mathcal{HB}_e$.

Proof. Based on Theorem 17 and Theorem 10,
M is a model of $Circum(T; P^1 > \ldots > P^k; Z)$
iff there is a consistent answer set S of Π_β such that $M = S \cap \mathcal{HB}_e$
iff there is a consistent stable argument extension $\mathcal{E}$ of $\mathbf{ABF}(\Pi_\beta)$ such that $S \cup \Delta_S = \mathrm{Concs}(\mathcal{E})$ and $M = S \cap \mathcal{HB}_e$
iff there is a consistent stable argument extension $\mathcal{E}$ of $\mathbf{ABF}(\Pi_\beta)$ such that $M = \mathrm{Concs}(\mathcal{E}) \cap \mathcal{HB}_e$ (due to $M = S \cap \mathcal{HB}_e = (S \cup \Delta_S) \cap \mathcal{HB}_e = \mathrm{Concs}(\mathcal{E}) \cap \mathcal{HB}_e$). □

Theorem 19 (resp. Theorem 17) indicates that reasoning in prioritized circumscription can be computed based on assumption-based argumentation (resp. answer set programming).

Example 13. Consider prioritized circumscription given in [35, Example 3.9]:

$$Circum\big(\{p \vee q, q \vee r\}; p > q; r\big).$$

It has two models $M_1 = \{\neg p, q, r\}$ and $M_2 = \{\neg p, q, \neg r\}$. Based on Theorem 17, these models can be obtained from answer sets of the EDLP $\Pi_\beta = (\Pi_\alpha)_1 \cup (\Pi_\alpha)_2 \cup \Pi_\gamma$ translated from it. In this case,

$$
\begin{aligned}
&(\Pi_\alpha)^1 = \{q1 \leftarrow not\ p1,\quad q1 | \, r1 \leftarrow,\quad \neg p1 \leftarrow not\ p1,\quad q1 | \neg q1 \leftarrow,\quad r1 | \neg r1 \leftarrow\},\\
&(\Pi_\alpha)^2 = \{p2 \leftarrow not\ q2,\quad r2 \leftarrow not\ q2,\quad p2 | \, q2 \leftarrow,\quad \neg q2 \leftarrow not\ q2,\quad r2 | \neg r2 \leftarrow,\quad p2 | \neg p2 \leftarrow\},\\
&\Pi_\gamma = \{p \leftarrow p1,\quad q \leftarrow q1,\quad r \leftarrow r1,\quad \neg p \leftarrow \neg p1,\quad \neg q \leftarrow \neg q1,\quad \neg r \leftarrow \neg r1,\\
&\qquad\ \ p \leftarrow p2,\quad q \leftarrow q2,\quad r \leftarrow r2,\quad \neg p \leftarrow \neg p2,\quad \neg q \leftarrow \neg q2,\quad \neg r \leftarrow \neg r2\}.
\end{aligned}
$$

where $\mathcal{HB}_e = \{p, q, r, \neg p, \neg q, \neg r\}$. Then Π_β has two answer sets S_1 and S_2 [35]:

$$S_1 = \{\neg p, q, r, \neg p1, q1, r1, \neg p2, q2, r2\}, \qquad S_2 = \{\neg p, q, \neg r, \neg p1, q1, \neg r1, \neg p2, q2, \neg r2\},$$

where $\quad S_1 \cap \mathcal{HB}_e = \{\neg p, q, r\} = M_1, \qquad S_2 \cap \mathcal{HB}_e = \{\neg p, q, \neg r\} = M_2.$

Thus, models of $Circum(\{p \vee q, q \vee r\}; p > q; r)$ are computed based on answer set semantics.

On the other hand, based on Theorem 19, these models are also obtained in argumentation as follows.

Consider $\mathbf{ABF}(\Pi_\beta) = \langle \mathfrak{L}, \Pi_\beta, \mathcal{A}_{\Pi_\beta}, \overline{} \rangle$ translated from Π_β. It has arguments and *attacks* as follows:

$$
\begin{aligned}
&A_1 : \{not\ p1\} \vdash q1,\quad A_2 : \{not\ r1\} \vdash q1,\quad A_3 : \{not\ q1\} \vdash r1,\quad A_4 : \{not\ p1\} \vdash \neg p1,\\
&A_5 : \{not\ q1\} \vdash \neg q1,\quad A_6 : \{not\ \neg q1\} \vdash q1,\quad A_7 : \{not\ r1\} \vdash \neg r1,\\
&A_8 : \{not\ \neg r1\} \vdash r1,\quad A_9 : \{not\ q2\} \vdash p2,\quad A_{10} : \{not\ q2\} \vdash r2,\\
&A_{11} : \{not\ p2\} \vdash q2,\quad A_{12} : \{not\ q2\} \vdash \neg q2,\quad A_{13} : \{not\ r2\} \vdash \neg r2,\\
&A_{14} : \{not\ \neg r2\} \vdash r2,\quad A_{15} : \{not\ \neg p2\} \vdash p2,\quad A_{16} : \{not\ p2\} \vdash \neg p2,\\
&A_{17} : \{not\ q2\} \vdash p,\quad A_{18} : \{not\ \neg p2\} \vdash p,\quad A_{19} : \{not\ p1\} \vdash q,\\
&A_{20} : \{not\ r1\} \vdash q,\quad A_{21} : \{not\ \neg q1\} \vdash q,\quad A_{22} : \{not\ p2\} \vdash q,\quad A_{23} : \{not\ q1\} \vdash r,\\
&A_{24} : \{not\ \neg r1\} \vdash r,\quad A_{25} : \{not\ q2\} \vdash r,\quad A_{26} : \{not\ \neg r2\} \vdash r,\quad A_{27} : \{not\ p1\} \vdash \neg p,\\
&A_{28} : \{not\ p2\} \vdash \neg p,\quad A_{29} : \{not\ q1\} \vdash \neg q,\quad A_{30} : \{not\ q2\} \vdash \neg q,\\
&A_{31} : \{not\ r1\} \vdash \neg r,\quad A_{32} : \{not\ r2\} \vdash \neg r,\\
&A_{\ell_i} : \{not\ x\} \vdash not\ x,\quad \text{where } x \in \{p, q, r, \neg p, \neg q, \neg r\} \text{ for } 33 \leqslant \ell_i \leqslant 38,\\
&A_{m_i} : \{not\ x1\} \vdash not\ x1,\quad \text{where } x1 \in \{p1, q1, r1, \neg p1, \neg q1, \neg r1\} \text{ for } 39 \leqslant m_i \leqslant 44,\\
&A_{n_i} : \{not\ x2\} \vdash not\ x2,\quad \text{where } x2 \in \{p2, q2, r2, \neg p2, \neg q2, \neg r2\} \text{ for } 45 \leqslant n_i \leqslant 50), \quad \text{and}
\end{aligned}
$$

$$
\begin{aligned}
attacks = \big\{&(A_1, A_3), (A_1, A_5), (A_1, A_{23}), (A_1, A_{29}), (A_1, A_{40}), (A_2, A_3), (A_2, A_5), (A_2, A_{23}),\\
&(A_2, A_{29}), (A_2, A_{40}), (A_3, A_2), (A_3, A_7), (A_3, A_{20}), (A_3, A_{31}), (A_3, A_{41}), (A_4, A_{42}),\\
&(A_5, A_6), (A_5, A_{21}), (A_5, A_{43}), (A_6, A_3), (A_6, A_5), (A_6, A_{23}), (A_6, A_{29}), (A_6, A_{40}),\\
&(A_7, A_8), (A_7, A_{24}), (A_7, A_{44}), (A_8, A_2), (A_8, A_7), (A_8, A_{20}), (A_8, A_{31}), (A_8, A_{41}),\\
&(A_9, A_{11}), (A_9, A_{16}), (A_9, A_{22}), (A_9, A_{28}), (A_9, A_{45}), (A_{10}, A_{13}), (A_{10}, A_{32}), (A_{10}, A_{50}),\\
&(A_{11}, A_9), (A_{11}, A_{10}), (A_{11}, A_{12}), (A_{11}, A_{17}), (A_{11}, A_{25}), (A_{11}, A_{30}), (A_{11}, A_{46}),\\
&(A_{12}, A_{49}), (A_{13}, A_{14}), (A_{13}, A_{26}), (A_{13}, A_{50}), (A_{14}, A_{13}), (A_{14}, A_{32}), (A_{14}, A_{50}),\\
&(A_{15}, A_{11}), (A_{15}, A_{16}), (A_{15}, A_{22}), (A_{15}, A_{28}), (A_{15}, A_{45}), (A_{16}, A_{15}), (A_{16}, A_{18}),\\
&(A_{16}, A_{48}), (A_{17}, A_{33}), (A_{18}, A_{33}), (A_{19}, A_{34}), (A_{20}, A_{34}), (A_{21}, A_{34}), (A_{22}, A_{34}),\\
&(A_{23}, A_{35}), (A_{24}, A_{35}), (A_{25}, A_{35}), (A_{26}, A_{35}), (A_{27}, A_{36}), (A_{28}, A_{36}), (A_{29}, A_{37}),\\
&(A_{30}, A_{37}), (A_{31}, A_{38}), (A_{32}, A_{38})\big\}.
\end{aligned}
$$

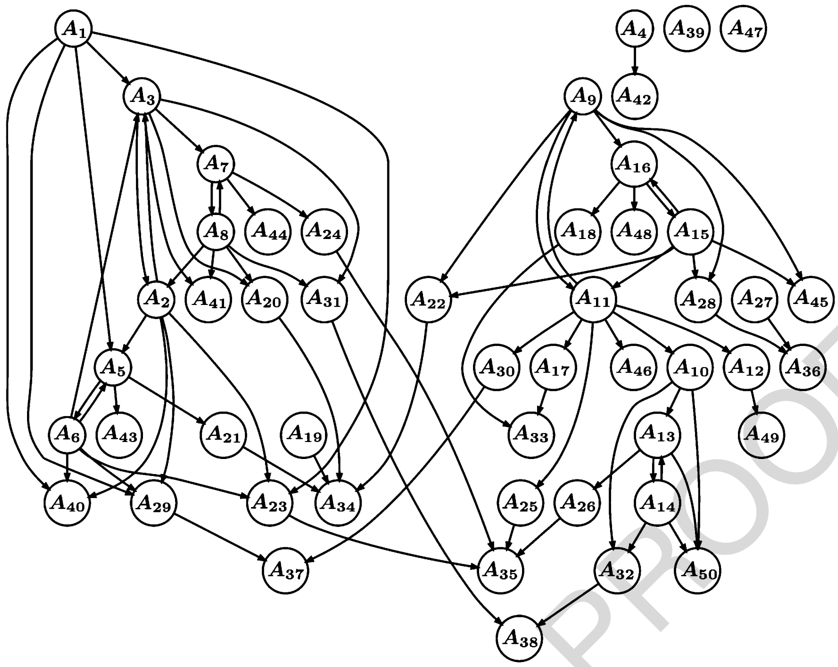

Fig. 5. The graphic representation of arguments and *attacks* in $\mathbf{ABF}(\Pi_\beta)$.

Fig. 5 shows the graphic representation of the AA framework generated from $\mathbf{ABF}(\Pi_\beta)$.

$\mathbf{ABF}(\Pi_\beta)$ has six stable argument extensions $\mathcal{E}_i$ ($1 \leqslant i \leqslant 6$), in which both $\mathcal{E}_1$ and $\mathcal{E}_2$ shown below are consistent, but the others are inconsistent:

$$\mathcal{E}_1 = \{A_1, A_4, A_6, A_8, A_{11}, A_{14}, A_{16}, A_{19}, A_{21}, A_{22}, A_{24}, A_{26}, A_{27}, A_{28}, A_{33}, A_{37}, A_{38}, A_{39}, A_{43}, A_{44}, A_{45}, A_{47}, A_{49}\},$$

$$\mathcal{E}_2 = \{A_1, A_2, A_4, A_6, A_7, A_{11}, A_{13}, A_{16}, A_{19}, A_{20}, A_{21}, A_{22}, A_{27}, A_{28}, A_{31}, A_{32}, A_{33}, A_{35}, A_{37}, A_{39}, A_{41}, A_{43}, A_{45}, A_{47}, A_{49}\},$$

where $\mathrm{Concs}(\mathcal{E}_1) \cap \mathcal{HB}_e = \{\neg p, q, r\} = M_1$, $\mathrm{Concs}(\mathcal{E}_2) \cap \mathcal{HB}_e = \{\neg p, q, \neg r\} = M_2$. Thus, models of $\mathit{Circum}(\{p \vee q, q \vee r\}; p > q; r)$ are computed based on assumption-based argumentation.

5. Relation to the possible model semantics of EDLPs

Minimality-based semantics interprets disjunctions *as exclusive as possible* as addressed in [30]. Instead, to freely specify both inclusive and exclusive interpretations of disjunctions, the possible model semantics of an EDLP was introduced by Sakama and Inoue [30].

In this paper, we restrict our attention to a *consistent* possible model which does not contain a pair of complementary literals L and $\neg L$. In the possible model semantics, a *split program*[14] of an EDLP P is

[14]A split program of an EDLP can be encoded by *choice rules* supported in recent ASP solvers (e.g. clingo) [21].

defined as a ground ELP obtained from P by replacing every ground disjunctive rule $r: L_1|\cdots|L_\ell \leftarrow \Gamma$ from P with the nondisjunctive rules in $R \subseteq Split_r$, where

$$Split_r = \{L_i \leftarrow \Gamma \mid \Gamma = body(r) \text{ for } r \in P,\ i = 1, \ldots, \ell\}$$

and R is a non-empty subset of $Split_r$. Each rule in $Split_r$ is called a *split rule* of r. We denote by $sp(P)$ a split program of an EDLP P as defined as follows: given an EDLP $P = \bigcup_j \{r_j\}$, where r_j is a rule,

$$sp(P) = \bigcup_j R_{r_j}, \text{ where each } r_j \text{ is replaced with } R_{r_j} \text{ s.t. } R_{r_j} \subseteq Split_{r_j} \text{ and } R_{r_j} \neq \emptyset.$$

P has multiple split programs in general. Then, a *consistent possible model* of P is defined as a consistent answer set of a split program of P. It should be noticed that a possible model of P is not always minimal among possible models of P. Any consistent answer set of P is a consistent possible model of P but not vice versa [30]. An EDLP is *consistent* under the possible model semantics iff it has a consistent possible model; otherwise it is inconsistent.

The possible models of an EDLP can be captured by stable extensions of a standard ABA framework since each of its split programs is an ELP as follows.

Theorem 20. *Let P be an EDLP and $sp(P)$ a split program of P. Then S is a consistent possible model of P iff there is a split program $sp(P)$ which has a consistent stable argument extension $\mathcal{E}$ of the ABA framework $\mathcal{F}(sp(P)) = \langle \mathcal{L}_P, sp(P), \mathcal{A}_P, \overline{} \rangle$ such that $S \cup \Delta_S = \mathrm{Concs}(\mathcal{E})$ (in other words, $S = \mathrm{Concs}(\mathcal{E}) \cap Lit_P$), where Δ_S is a consistent stable assumption extension of the ABF $\mathcal{F}(sp(P))$.*

Proof. S is a consistent possible model of P
iff there is a split program $sp(P)$ which has a consistent answer set S
(due to the definition of a consistent possible model)
iff there is a consistent stable argument extension $\mathcal{E}$ of the ABA framework $\mathcal{F}(sp(P))$ such that $S \cup \Delta_S = \mathrm{Concs}(\mathcal{E})$, (due to Theorem 4)
where Δ_S is a consistent stable assumption extension of the ABF $\mathcal{F}(sp(P))$ due to Proposition 14. $\square$

In what follows, we define an *argument possible-extension* (an *argument p-extension*, for short) and an *assumption possible-extension* (an *assumption p-extension*, for short) of $\mathbf{ABF}(P)$ which introduces the idea of possible models of an EDLP P in ABA.

Definition 24. Let P be an EDLP, $sp(P)$ a split program of P and $\mathbf{ABF}(sp(P)) = \langle \mathfrak{L}, sp(P), \mathcal{A}_P, \overline{} \rangle$. We say that $\mathcal{E}$ is a *stable argument possible-extension* (or a stable argument p-extension, for short) of $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{} \rangle$ if $\mathcal{E}$ is a stable argument extension of $\mathbf{ABF}(sp(P))$ for some ELP $sp(P)$; and $\mathcal{E}$ is a *consistent* stable argument possible-extension (or a *consistent* stable argument p-extension) of $\mathbf{ABF}(P)$ if $\mathcal{E}$ is a *consistent* stable argument extension of $\mathbf{ABF}(sp(P))$ for some ELP $sp(P)$.

Similarly, we say that Δ is a *stable assumption possible-extension* (or a stable assumption p-extension, for short) of $\mathbf{ABF}(P)$ if Δ is a stable assumption extension of $\mathbf{ABF}(sp(P))$ for some ELP $sp(P)$; and Δ is a *consistent* stable assumption possible-extension (or a *consistent* stable assumption p-extension) of $\mathbf{ABF}(P)$ if Δ is a *consistent* stable assumption extension of $\mathbf{ABF}(sp(P))$ for some ELP $sp(P)$.

Notice that only the inference rule [MP] is needed to compute p-extensions of $\mathbf{ABF}(P)$ because any $sp(P)$ contains no disjunction. Regarding p-extensions, the following proposition and theorem hold.

Proposition 11. *Let* $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{\ }\rangle$ *be the ABF translated from an EDLP* P. *Then* S *is a consistent possible model of* P *iff there is a consistent stable assumption p-extension* Δ_S *of* $\mathbf{ABF}(P)$ *such that* $\Delta_S = \{not\ \ell \mid \ell \in (Lit_P \setminus S)\}$ *where* $\mathsf{CN}_{sp(P)}(\Delta_S) = S \cup \Delta_S$ *for some split program* $sp(P)$.

Proof. S is a consistent possible model of P
iff there is a split program $sp(P)$ which has a consistent answer set S (due to its definition)
iff there is a consistent stable assumption extension Δ_S of $\mathbf{ABF}(sp(P))$ for a split program $sp(P)$ s.t. $\Delta_S = \{not\ \ell \mid \ell \in (Lit_P \setminus S)\}$ where $\mathsf{CN}_{sp(P)}(\Delta_S) = S \cup \Delta_S$ is consistent (due to Proposition 9)
iff there is a consistent stable assumption p-extension Δ_S of $\mathbf{ABF}(P)$ such that $\Delta_S = \{not\ \ell \mid \ell \in (Lit_P \setminus S)\}$ where $\mathsf{CN}_{sp(P)}(\Delta_S) = S \cup \Delta_S$ for some $sp(P)$. □

Theorem 21. *Let* $\mathbf{ABF}(P) = \langle \mathfrak{L}, P, \mathcal{A}_P, \overline{\ }\rangle$ *be the ABF translated from an EDLP* P. S *is a consistent possible model of* P *iff there is a consistent stable argument p-extension* $\mathcal{E}$ *of* $\mathbf{ABF}(P)$ *such that* $S \cup \Delta_S = \mathrm{Concs}(\mathcal{E})$, *where* $\Delta_S = \{not\ \ell \mid \ell \in (Lit_P \setminus S)\}$ *is a consistent stable assumption p-extension of* $\mathbf{ABF}(P)$.

Proof. S is a consistent possible model of P
iff there is a consistent stable argument extension $\mathcal{E}$ along with the consistent stable assumption extension $\Delta_S = \{not\ \ell \mid \ell \in (Lit_P \setminus S)\}$ of $\mathbf{ABF}(sp(P))$ for some split program $sp(P)$ such that $\mathrm{Concs}(\mathcal{E}) = S \cup \Delta_S = \mathsf{CN}_{sp(P)}(\Delta_S)$ which is consistent (due to Proposition 11 and Theorem 10)
iff there is a consistent stable argument p-extension $\mathcal{E}$ of $\mathbf{ABF}(P)$ such that $S \cup \Delta_S = \mathrm{Concs}(\mathcal{E})$, where $\Delta_S = \{not\ \ell \mid \ell \in (Lit_P \setminus S)\}$ is a consistent stable assumption p-extension of $\mathbf{ABF}(P)$. □

For an EDLP P with no disjunction, Proposition 11 reduces to Proposition 14, while Theorem 21 reduces to Theorem 4 respectively.

The following example shows that there is a case that needs the inclusive interpretation of disjunction though Kyoto protocol problem in Section 1 needs the exclusive interpretation of the disjunction $p \mid y$.

Example 14 (**Taxi fare problem**). Consider the following problem.

Each of Jack and Mary takes a taxi if it is available. Suppose that a taxi is available, where its fare is v dollars. If they share it, each may pay only the half of v (say hv) for a taxi fare; otherwise anyone who takes it should pay v dollars.

This situation can be represented by the following EDLP P_5, where j (resp. m) denotes that Jack (resp. Mary) takes a taxi, tx denotes that a taxi is available, and v (resp. hv) denotes that a person who takes a taxi pays v (resp. hv) dollars:

$$P_5 = \{j \mid m \leftarrow tx, \quad tx \leftarrow, \quad hv \leftarrow j, m, \quad v \leftarrow j, not\ m, \quad v \leftarrow m, not\ j\}.$$

P_5 has two answer sets $S_1 = \{tx, j, v\}$ and $S_2 = \{tx, m, v\}$, where S_1 (resp. S_2) shows the case that *Jack* (resp. *Mary*) pays v dollar due to taking a taxi alone.

On the other hand, P_5 has three split programs: $\pi_1 = \pi \cup \{j \leftarrow tx\}$, $\pi_2 = \pi \cup \{m \leftarrow tx\}$ and $\pi_3 = \pi \cup \{j \leftarrow tx, \quad m \leftarrow tx\}$, where $\pi = \{v \leftarrow j, not\ m, \quad v \leftarrow m, not\ j, \quad hv \leftarrow j, m, \quad tx \leftarrow\}$. Since each π_i has the unique answer set S_i $(1 \leqslant i \leqslant 3)$, P_5 has three possible models S_1, S_2 along with $S_3 = \{tx, j, m, hv\}$ which corresponds to the third case such that both of *Jack* and *Mary* pay *hv* dollars thanks to sharing a taxi. Notice that S_3 is not the answer set of P_5.[15]

[15] However, when the problem is expressed by the EDLP $P_5' = P_5 \cup \{j|\neg j \leftarrow, \quad m|\neg m \leftarrow\}$ instead of P_5, P_5' has three answer sets, $S_1' = \{tx, j, v, \neg m\}$, $S_2' = \{tx, m, v, \neg j\}$ and $S_3 = \{tx, j, m, hv\}$ corresponding to three cases.

In contrast, the same goes for argumentation under the stable semantics. First we construct $\mathbf{ABF}(P_5)$ translated from P_5 (resp. $\mathbf{ABF}(\pi_i)$ translated from π_i $(1 \leqslant i \leqslant 3)$). In this case, every extension of the respective ABF is consistent since each language does not include explicit negation. Then, $\mathbf{ABF}(\pi_1)$ (resp. $\mathbf{ABF}(\pi_2)$, $\mathbf{ABF}(\pi_3)$) has the unique stable assumption extension $\Delta_1 = \{not\ m,\ not\ hv\}$ (resp. $\Delta_2 = \{not\ j,\ not\ hv\}$, $\Delta_3 = \{not\ v\}$), where $\mathsf{CN}_{\pi_1}(\Delta_1) = \mathsf{CN}_{P_5}(\Delta_1) = \{tx, j, v\} \cup \Delta_1 = S_1 \cup \Delta_1$, $\mathsf{CN}_{\pi_2}(\Delta_2) = \mathsf{CN}_{P_5}(\Delta_2) = \{tx, m, v\} \cup \Delta_2 = S_2 \cup \Delta_2$ and $\mathsf{CN}_{\pi_3}(\Delta_3) = \{tx, j, m, hv\} \cup \Delta_3 = S_3 \cup \Delta_3 \neq \mathsf{CN}_{P_5}(\Delta_3)$. Similarly, $\mathbf{ABF}(\pi_1)$ (resp. $\mathbf{ABF}(\pi_2)$, $\mathbf{ABF}(\pi_3)$) has the unique stable argument extension $\mathcal{E}_1$ (resp. $\mathcal{E}_2$, $\mathcal{E}_3$), where $\mathrm{Concs}(\mathcal{E}_i) = S_i \cup \Delta_i$ $(1 \leqslant i \leqslant 3)$. Hence $\mathbf{ABF}(P_5)$ has three stable assumption p-extensions Δ_i $(1 \leqslant i \leqslant 3)$ along with three argument p-extensions $\mathcal{E}_i$ $(1 \leqslant i \leqslant 3)$, while it has two stable assumption extensions Δ_1, Δ_2 along with two argument extensions $\mathcal{E}_1$, $\mathcal{E}_2$.

6. Related work

Beirlaen et al.'s extended ASPIC$^+$ framework [2,3] as well as Heyninck and Arieli's ABF induced by a DLP [18] can handle disjunctive information in argumentation. Moreover, it is shown that Heyninck and Arieli's ABF induced by a DLP can capture stable model semantics of a DLP as shown in Proposition 1 [18, Proposition 2 and Proposition 3]. However the semantic correspondence to a disjunctive default theory is not shown in both Beirlaen et al.'s work and Heyninck and Arieli's work. Notice here that Example 4 illustrates that an NDP (i.e. an EDLP having no classical negation) can be transformed to a disjunctive default theory, whereas a DLP used in Heyninck and Arieli's ABF can neither be done so nor has the semantic correspondence to a default theory. In contrast, our approach based on a translation from an EDLP can capture not only the answer set semantics of an EDLP but also the semantics of a disjunctive default theory as shown in Theorem 8 (or Proposition 5) and Theorem 12. These are written in Table 1 and Table 2 in which existing work is shown in italic letter with proper references. Regarding the problem of how to construct arguments under the given disjunctive information, Beirlaen et al. [2,3] allowed arguments to have disjunctive conclusions, while Heyninck and Arieli [18] neither defined arguments nor took account of argument extensions in their ABFs. In contrast, in our ABFs, arguments are defined as trees due to Definition 12 but they are not allowed to have disjunctive conclusions according to Definition 13. Notice here that Heyninck and Arieli's ABF has also three inference rules: [MP], [Res] and [RBC] like our ABFs though the classical $\vee$ is used instead of $|$. Then, thanks to our approach, it is possible to construct trees and define arguments in their ABF based on the slightly modified Definition 12 by replacing $|$ with $\vee$. As a result, we may obtain a new theorem for a DLP P showing the correspondence between stable models of P and stable argument extensions of their $\mathbf{ABF}(P)$. Though it is possible to fill in the open box of Table 1 with such a theorem for a DLP, its details are omitted in this paper.

Lehtonen et al. [19] presented algorithms for reasoning in a default logic instantiation of ABA, where they defined the assumption-based argumentation framework (ABF) corresponding to a propositional default theory. In [19, Example 1], they showed the ABF corresponding a default theory which contains disjunctive formulas. However when the ABF corresponding to the default theory D_1 is constructed according to [19, Definition 1], the ABF has the same difficulty as D_1 addressed in Example 1. The reason is as follows: In their approach, given $(\mathcal{L}_p, \mathcal{R}_p)$ as a deductive system for propositional logic, the ABF corresponding to D_1 is $F = (\mathcal{L}, \mathcal{R}, W, \mathcal{A}, \bar{\ })$, where $\mathcal{L} = \mathcal{L}_p \cup \{M\alpha \mid \alpha \in \mathcal{L}_p\}$, $\mathcal{R} = \mathcal{R}_p \cup \{r \leftarrow p, Mr,\quad k \leftarrow r, Mk,\quad f \leftarrow y, Mf,\quad k \leftarrow f, Mk\}$, $W = D_1 \cap \mathcal{L}_p = \{p \vee y\}$, $\mathcal{A} = \{Mr, Mk, Mf\}$ and a mapping function $\bar{\ }$ defined by $\overline{Mx} = \neg x$ for all $Mx \in \mathcal{A}$. Besides, $A \subseteq \mathcal{A}$ *attacks* $B \subseteq \mathcal{A}$ iff

Table 1

Correspondence between LP and ABA

LP		ABA	
Class	Semantics	Assumption extensions	Argument extensions
ELP	p-stable models	Proposition 12	*Theorem 2* [32]
	answer sets	Proposition 13 Proposition 14	*Theorem 3* [32] *Theorem 4* [33]
DLP	stable models	*Proposition 1* [18]	
NDP	answer sets	Proposition 3	Theorem 6
EDLP	p-stable models	Proposition 4	Theorem 7
	answer sets	Proposition 5 Proposition 9	Theorem 8 Theorem 10
	possible models	Proposition 11	Theorem 21

Table 2

Correspondence between NMR and ABA

NMR		EDLP	ABA
Formalisms	Semantics	Answer sets	Argument extensions
disjunctive default theory	extensions	*Theorem 1* [16]	Theorem 12 Theorem 13
parallel circumscription	models	*Theorem 16* [29]	Theorem 18
prioritized circumscription	models	Theorem 17	Theorem 19

$W \cup A \vdash_{\mathcal{R}} \neg b$ for some $Mb \in B$, where $\vdash_{\mathcal{R}}$ denotes derivability via $\mathcal{R}$. Then since there is no *attacks* between any two sets of assumptions, the ABF F has a unique assumption extension $\mathcal{A}$ under the stable (resp. grounded, complete) semantics, from which the expected result k is never derived under each semantics due to [19, Proposition 1]. In contrast, in our approach based on the disjunctive default theory D_2 which has the associated EDLP P_1, Example 12 shows that we can obtain the expected result k from the stable extensions $\mathcal{E}_i$ of the ABF translated from the EDLP P_1 which correspond to the extensions $E_i (i = 1, 2)$ of the *ddt* $D_2 = emb(P_1)$ based on Theorem 12, Theorem 8 and Proposition 5. Moreover, interestingly, Example 8 shows that the skeptical result meant by the problem is obtained under the skeptical semantics (i.e. the grounded and ideal semantics) in our ABF.

Bondarenko et al. [5] showed a correspondence between Reiter's default extensions and stable extensions of the corresponding assumption-based framework (ABF) in [5, Theorem 3.16]. This indicates that *Kyoto protocol problem* shown in Example 1 cannot be solved under the stable semantics in their ABF corresponding to the default theory D_1. Disjunctive default theories are not considered in [5]. Moreover, in [5, Theorem 6.7], Bondarenko et al. showed a correspondence between models of parallel circumscription:$Circum(T; P; Z)$ (i.e. P, Z-minimal models of T) and maximal conflict-free extensions of the corresponding ABF, assuming that T is a first order theory without function symbols, T satisfies unique names axioms and domain closure axioms, and every model of T is a Herbrand model of T. However, they showed nothing about a semantic correspondence between prioritized circumscription and assumption-based frameworks. In contrast, this paper shows a correspondence between Herbrand models of parallel circumscription and stable *argument* extensions of our ABF in Theorem 18 as well as a correspondence between Herbrand models of prioritized circumscription and stable *argument* extensions of our ABF in Theorem 19 (see Table 2). Regarding logic programming, they showed a correspondence between stable models of an NLP and the stable extensions of the corresponding ABF in [5, Theorem 3.13]. Their theorem is, however, the special case of Proposition 13 for an ELP, Proposition 1 for a DLP, Proposition 3 for an NDP, and Proposition 5 for an EDLP.

Caminada and Schulz [7] showed the equivalence between various ABA semantics and various semantics of NLPs. NLPs prohibiting both disjunction and classical negation are less expressive than DLPs and ELPs. Hence a faithful modeling of real world problems often becomes impossible in the scope of NLPs.

7. Conclusion

We proposed an assumption-based framework (ABF) translated from an extended disjunctive logic program (EDLP), which incorporates explicit negation as well as $|$ rather than $\vee$ in Heyninck and Arieli's ABF induced by a DLP. Thanks to our proposed ABFs, the new results about the semantic relationships between logic programming (LP) and ABA as well as nonmonotonic reasoning (NMR) and ABA are obtained. That is, as for LP, the answer set semantics of an EDLP can be captured by the stable extensions of the ABF translated from an EDLP with trivialization rules, while as for NMR, extensions of a disjunctive default theory (resp. models of prioritized circumscription) can be captured by the stable extensions of the ABF translated from the EDLP corresponding to a given disjunctive default theory (resp. the EDLP corresponding to a given prioritized circumscription). Moreover, as another relationship to LP, it is shown that the possible model semantics of an EDLP is captured by the possible extensions under stable semantics of the ABF translated from an EDLP (see Table 1 and Table 2).

In the study of nonmonotonic reasoning, disjunctive default logic [16] was proposed as a generalization of default logic [27] to overcome difficulties of default logic in handling disjunctive information. In fact, defaults *do not work* in the default theory D_1 expressing the Kyoto protocol problem [3] as well as in d_1 corresponding to the DLP π_1 [18, Example 1] shown in Example 4 due to lack of the capability to reason by cases in these default theories. As a result, D_1 as well as the ABFs instantiated with D_1 [5,19] reveal difficulties such that the expected result cannot be obtained from them as discussed in Section 1 and Section 6. In contrast, in our approach, the expected result k is successfully obtained not only from the stable extensions of the ABF translated from the EDLP P_1 which correspond to the extensions of the disjunctive default theory $D_2 = emb(P_1)$ as shown in Example 12 but also from the extension(s) under the skeptical semantics (i.e. the grounded and ideal) as well as under the preferred semantics in our proposed ABF as shown in Example 8. Thus based on our ABFs corresponding to disjunctive default theories via EDLPs, our approach can overcome difficulties of assumption-based frameworks corresponding to default theories (e.g. [5,19]) in handling disjunctive information.

To sum up, as for argument extensions, Theorem 2 [32, Theorem 3] and Theorem 3 [32, Theorem 4] for an ELP (resp. Theorem 4 [33, Theorem 5] for a consistent ELP) in standard ABA frameworks are broadened to Theorem 7 and Theorem 8 for an EDLP (resp. Theorem 10 for a consistent EDLP) in generalized ABA frameworks, i.e. ABFs translated from EDLPs. Similarly as for assumption extensions, Proposition 12 and Proposition 13 (resp. Proposition 14) for the standard ABA framework instantiated with an ELP (resp. a consistent ELP) as well as Proposition 1 [18, Proposition 2 and Proposition 3] for the ABF induced by a DLP are generalized to Proposition 4 and Proposition 5 (resp. Proposition 9) for the respective ABFs translated from EDLPs (resp. a consistent EDLP). These are summarized in Table 1.

As one of practical advantages of our approach, even if disjunctive information exists, we can directly use dialectic proof procedures [9,11] since the AA framework [8] can be generated from our ABF treating disjunctive information.

In (extended) disjunctive logic programming, the existence of disjunction generally increases the expressive power of logic programs while brings computational penalty [13]. By analogy, argumentation in ABFs translated from (E)DLPs increases the expressive power of ABF while it would introduce additional complexity. Hence, the analysis of complexity is left for future work. Moreover, our future work is to explore and find the more general correspondence between Assumption-based frameworks and disjunctive default theories without intervening EDLPs.

Acknowledgements

The author would like to thank Chiaki Sakama and the anonymous reviewers of the paper for their valuable comments and suggestions.

Appendix. New theorems for an ELP

A.1. Correspondence between answer sets of an ELP and stable assumption extensions

We show the correspondence between p-stable models (resp. answer sets) of an ELP P and stable *assumption* extensions in the standard ABA framework $\mathcal{F}(P)$ (resp. $\mathcal{F}(P_{tr})$) as follows.

Lemma 6. *Let P be an ELP and $M \subseteq Lit_P$. We denote by $\vdash$ derivability using modus ponens. Then M is a p-stable model of an ELP P iff $M = \{\ell \in Lit_P \mid P \cup \Delta_M \vdash \ell\}$, where $\Delta_M = \{not\ \ell \mid \ell \in (Lit_P \setminus M)\}$.*

Proof. It is proved in [5, p.80] using the inference rule (i.e. modus ponens) in [5, p.72, line 1] that

$$M \text{ is a stable model of an NLP } P \text{ iff } M = \{p \in HB_P \mid P \cup \Delta_M \vdash p\}$$
$$\text{where } \Delta_M = \{not\ p \mid p \notin M\}. \tag{13}$$

Now let P^+ be the NLP which is the positive form [28] of an ELP P, and for a set $S \subseteq Lit_P$, let S^+ be the set obtained by replacing each negative literal $\neg L$ in S with a newly introduced atom L' like in the proof of Proposition 4. Then the Herbrand base HB_{P^+} of P^+ is $(Lit_P)^+$.

(i) Suppose that M is a p-stable model of an ELP P. Then M^+ is an answer set of the NLP P^+ due to Lemma 3. Hence M^+ is a stable model of the NLP P^+ due to Lemma 1. Then based on (13), it holds that $M^+ = \{p \in HB_{P^+} \mid P^+ \cup \Delta_{M^+} \vdash p\}$ where $\Delta_{M^+} = \{not\ p \mid p \notin M^+\} = \{not\ p \mid p \in (HB_{P^+} \setminus M^+)\}$, which leads to $M = \{\ell \in Lit_P \mid P \cup \Delta_M \vdash \ell\}$ where $\Delta_M = \{not\ \ell \mid \ell \in (Lit_P \setminus M)\}$.
(ii) The converse is also proved in a similar way to (i). $\square$

Proposition 12. *Let P be an ELP and $M \subseteq Lit_P$. If M is a p-stable model of P, then $\Delta = \{not\ \ell \mid \ell \in (Lit_P \setminus M\}$ is a stable assumption extension of the ABA framework $\mathcal{F}(P) = \langle \mathcal{L}_P, P, \mathcal{A}_P, \overline{} \rangle$. Conversely if $\Delta \subseteq \mathcal{A}_P$ is a stable assumption extension of the ABF $\mathcal{F}(P)$, then $M = \{\ell \in Lit_P \mid not\ \ell \notin \Delta\}$ is a p-stable model of P.*

Proof. Let M be a p-stable model of an ELP P. Due to Lemma 6, it holds that, for any $\ell \in Lit_p$, $\ell \in M$ iff $P \cup \Delta_M \vdash \ell$, where $\Delta_M = \{not\ \ell \mid \ell \in (Lit_P \setminus M)\} = \Delta$. This means that in the ABF $\mathcal{F}(P) = \langle \mathcal{L}_P, P, \mathcal{A}_P, \overline{} \rangle$, the set of assumption Δ_M attacks *not* ℓ where $\ell \in M$. Besides Δ_M implies $M = \{\ell \in Lit_P \mid not\ \ell \in (\mathcal{A}_P \setminus \Delta_M)\}$. Hence Δ_M attacks *not* ℓ where *not* $\ell \in (\mathcal{A}_P \setminus \Delta_M)$. Hence

$\Delta = \Delta_M$ is conflict-free and stable. Thus Δ is a stable assumption extension of the ABA framework $\mathcal{F}(P)$.

Conversely let $\Delta \subseteq \mathcal{A}_P$ be a stable assumption extension of the ABA framework $\mathcal{F}(P)$ where $\mathcal{A}_P = NAF_P$. Then since Δ is stable, Δ is conflict-free, and Δ attacks every *not* $\ell \in (\mathcal{A}_P \setminus \Delta)$, where $\ell \in Lit_P$. This means that, $P \cup \Delta \nvdash \ell$ for *not* $\ell \in \Delta$, and

$$P \cup \Delta \vdash \ell \quad \text{for } not\ \ell \in (\mathcal{A}_P \setminus \Delta). \tag{14}$$

Based on (14), let us define the set M such that $M = \{\ell \in Lit_P \mid P \cup \Delta \vdash \ell \text{ for } not\ \ell \notin \Delta\}$. Then,

$$\Delta = \big\{not\ \ell \mid \ell \in (Lit_P \setminus M)\big\} \tag{15}$$

is derived using M. Hence, based on Lemma 6, M is a p-stable model of P. Moreover, (15) implies $M = \{\ell \in Lit_P \mid not\ \ell \notin \Delta\}$. Thus $M = \{\ell \in Lit_P \mid not\ \ell \notin \Delta\}$ is a p-stable model of P. $\square$

Proposition 13. *Let P be an ELP and $S \subseteq Lit_P$. If S is an answer set of P, then $\Delta = \{not\ \ell \mid \ell \in (Lit_P \setminus S)\}$ is a stable assumption extension of the ABA framework $\mathcal{F}(P_{tr})$. Conversely if Δ is a stable assumption extension of the ABF $\mathcal{F}(P_{tr})$, then $S = \{\ell \in Lit_P \mid not\ \ell \notin \Delta\}$ is an answer set of P.*

Proof. (i) Suppose S is an answer set of an ELP P. Then due to [28, Theorem 3.5], S is a p-stable model of P_{tr}. Hence based on Proposition 12, $\Delta = \{not\ \ell \mid \ell \in (Lit_P \setminus S)\}$ is a stable assumption extension of the ABF $\mathcal{F}(P_{tr})$. (ii) The converse is also proved in a similar way. $\square$

For a consistent ELP, the following proposition holds.

Proposition 14. *Let P be a consistent ELP and $S \subseteq Lit_P$. S is an answer set of P iff there is a consistent stable assumption extension $\Delta = \{not\ \ell \mid \ell \in (Lit_P \setminus S)\}$ of the ABA framework $\mathcal{F}(P)$.*

Proof. For a p-stable model S of an ELP P, it holds that, $S = \{\ell \in Lit_P | P \cup \Delta \vdash \ell\}$ for $\Delta = \{not\ \ell | \ell \in (Lit_P \setminus S)\}$ due to Lemma 6. Based on it, it is derived that,

$$CN_P(\Delta) = \{x \in \mathcal{L}_P | P \cup \Delta \vdash x\} = S \cup \Delta, \quad \text{for } \Delta = \big\{not\ \ell |\ \ell \in (Lit_P \setminus S)\big\}, \tag{16}$$

where CN_P is a consequence operator given in Definition 2. Then

S is an answer set of a consistent ELP P
iff S is a consistent answer set of P
iff S is a consistent p-stable model of P
iff there is a stable assumption extension $\Delta = \{not\ \ell |\ \ell \in (Lit_P \setminus S)\}$ of the ABF framework $\mathcal{F}(P)$, where due to (16), it holds that $CN_P(\Delta) = S \cup \Delta$, which is not contradictory, i.e. consistent according to Definition 2
iff there is a consistent stable assumption extension $\Delta = \{not\ \ell |\ \ell \in (Lit_P \setminus S)\}$ of the ABF $\mathcal{F}(P)$. $\square$

References

[1] O. Arieli, A. Borg, J. Heyninck and C. Straßer, Logic-based approaches to formal argumentation, *IfCoLog Journal of Logics and their Applications* **8**(6) (2021), 1793–1898.

[2] M. Beirlaen, J. Heyninck and C. Straßer, Reasoning by cases in structured argumentation, in: *Proceedings of the 2017 ACM Symposium on Applied Computing*, ACM, 2017, pp. 989–994.

[3] M. Beirlaen, J. Heyninck and C. Straßer, A critical assessment of Pollock's work on logic-based argumentation with suppositions, in: *Proceedings of the 17th International Workshop on Non-monotonic Reasoning (NMR-2018)*, 2018, pp. 63–72.

[4] R. Ben-Eliyahu and R. Dechter, Propositional semantics for disjunctive logic programs, *Annals of Mathematics and Artificial Intelligence* **12**(1–2) (1994), 53–87. doi:10.1007/BF01530761.

[5] A. Bondarenko, P.M. Dung, R.A. Kowalski and F. Toni, An abstract, argumentation-theoretic approach to default reasoning, *Artificial Intelligence* **93** (1997), 63–101. doi:10.1016/S0004-3702(97)00015-5.

[6] M. Caminada and L. Amgoud, On the evaluation of argumentation formalisms, *Artificial Intelligence* **171**(5–6) (2007), 286–310. doi:10.1016/j.artint.2007.02.003.

[7] M. Caminada and C. Schulz, On the equivalence between assumption-based argumentation and logic programming, *Journal of Artificial Intelligence Research* **60** (2017), 779–825. doi:10.1613/jair.5581.

[8] 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) (1995), 321–358. doi:10.1016/0004-3702(94)00041-X.

[9] P.M. Dung, R.A. Kowalski and F. Toni, Dialectic proof procedures for assumption-based, admissible argumentation, *Artificial Intelligence* **170**(2) (2006), 114–159. doi:10.1016/j.artint.2005.07.002.

[10] P.M. Dung, R.A. Kowalski and F. Toni, Assumption-based argumentation, in: *Argumentation in Artificial Intelligence*, Springer-Verlag, 2009, pp. 199–218. doi:10.1007/978-0-387-98197-0_10.

[11] P.M. Dung, P. Mancarella and F. Toni, Computing ideal sceptical argumentation, *Artificial Intelligence* **171** (2007), 642–674. doi:10.1016/j.artint.2007.05.003.

[12] P.M. Dung and P.M. Thang, Closure and consistency in logic-associated argumentation, *Journal of Artificial Intelligence Research* **49** (2014), 79–109. doi:10.1613/jair.4107.

[13] T. Eiter and G. Gottlob, Complexity results for disjunctive logic programming and application to nonmonotonic logics, in: *Logic Programming*, Proceedings of the 10th International Symposium (ILPS-1993), MIT Press, 1993, pp. 266–278.

[14] M. Gelfond and V. Lifschitz, The stable model semantics for logic programming, in: *Logic Programming*, Proceedings of the Fifth International Conference and Symposium (ICLP/SLP-1988), MIT Press, 1988, pp. 1070–1080.

[15] M. Gelfond and V. Lifschitz, Classical negation in logic programs and disjunctive databases, *New Generation Computing* **9** (1991), 365–385. doi:10.1007/BF03037169.

[16] M. Gelfond, V. Lifschitz, H. Przymusinska and M. Truszczynski, Disjunctive defaults, in: *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR-1991)*, 1991, pp. 230–237.

[17] J. Heyninck and O. Arieli, On the semantics of simple contrapositive assumption-based argumentation frameworks, in: *Proceedings of the Seventh International Conference on Computational Models of Argument (COMMA-2018)*, IOS Press, 2018, pp. 9–20.

[18] J. Heyninck and O. Arieli, An argumentative characterization of disjunctive logic programming, in: *Proceedings of the 19th EPIA Conference on Artificial Intelligence (EPIA-2019)*, 2019, pp. 526–538.

[19] T. Lehtonen, J.P. Wallner and M. Järvisalo, Algorithms for reasoning in a default logic instantiation of assumption-based argumentation, in: *Proceedings of the 9th International Conference on Computational Models of Argument (COMMA-2022)*, IOS Press, 2022, pp. 236–247.

[20] V. Lifschitz, Computing circumscription, in: *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-1985)*, 1985, pp. 121–127.

[21] V.W. Marek, I. Niemelä and M. Truszczynski, Logic programs with monotone abstract constraint atoms, *Theory and Practice of Logic Programming* **8**(2) (2008), 167–199. doi:10.1017/S147106840700302X.

[22] J. McCarthy, Circumscription – a form of non-monotonic reasoning, *Artificial Intelligence* **13** (1980), 27–39. doi:10.1016/0004-3702(80)90011-9.

[23] J. McCarthy, Applications of circumscription to formalizing commonsense knowledge, *Artificial Intelligence* **28** (1986), 89–116. doi:10.1016/0004-3702(86)90032-9.

[24] S. Modgil and H. Prakken, The ASPIC+ framework for structured argumentation: A tutorial, *Argument and Computation* **5**(3) (2014), 31–62. doi:10.1080/19462166.2013.869766.

[25] H. Prakken, An abstract framework for argumentation with structured arguments, *Argument and Computation* **1**(2) (2010), 93–124. doi:10.1080/19462160903564592.

[26] T.C. Przymusinski, Stable semantics for disjunctive programs, *New Generation Computing* **9** (1991), 401–424. doi:10.1007/BF03037171.

[27] R. Reiter, A logic for default reasoning, *Artificial Intelligence* **13** (1980), 81–132. doi:10.1016/0004-3702(80)90014-4.

[28] C. Sakama and K. Inoue, Paraconsistent stable semantics for extended disjunctive programs, *Journal of Logic and Computation* **5**(3) (1995), 265–285. doi:10.1093/logcom/5.3.265.
[29] C. Sakama and K. Inoue, Embedding circumscriptive theories in general disjunctive programs, in: *Proceedings of the Third International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-1995)*, 1995, pp. 344–357. doi:10.1007/3-540-59487-6_25.
[30] C. Sakama and K. Inoue, Abductive logic programming and disjunctive logic programming: their relationship and transferability, *Journal of Logic and Computation* **44**(1–3) (2000), 75–100. doi:10.1016/S0743-1066(99)00073-4.
[31] C. Schulz and F. Toni, Justifying answer sets using argumentation, *Theory and Practice of Logic Programming* **16**(1) (2016), 59–110. doi:10.1017/S1471068414000702.
[32] T. Wakaki, Assumption-based argumentation equipped with preferences and its application to decision-making, practical reasoning, and epistemic reasoning, *Journal of Computational Intelligence* **33**(4) (2017), 706–736. doi:10.1111/coin.12111.
[33] T. Wakaki, Consistency in assumption-based argumentation, in: *Proceedings of the 8th International Conference on Computational Models of Argument (COMMA-2020)*, IOS Press, 2020, pp. 371–382.
[34] T. Wakaki, Assumption-based argumentation for extended disjunctive logic programming, in: *Proceedings of the 12th International Symposium on Foundations of Information and Knowledge Systems (FoIKS-2022)*, 2022, pp. 35–54.
[35] T. Wakaki and K. Satoh, Compiling prioritized circumscription into extended logic programs, in: *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-1997)*, 1997, pp. 182–187.
[36] https://arxiv.org/pdf/2106.10966.pdf.