Interlinking Logic Programs and Argumentation Frameworks

Chiaki Sakama[1(✉)] and Tran Cao Son[2]

[1] Wakayama University, 930 Sakaedani, Wakayama 640-8510, Japan
sakama@wakayama-u.ac.jp

[2] New Mexico State University, Las Cruces, NM 88003, USA
tson@cs.nmsu.edu

Abstract. *Logic programs* (LPs) and *argumentation frameworks* (AFs) are two declarative knowledge representation (KR) formalisms used for different reasoning tasks. The purpose of this study is interlinking two different reasoning components. To this end, we introduce two frameworks: *LPAF* and *AFLP*. The former enables to use the result of argumentation in AF for reasoning in LP, while the latter enables to use the result of reasoning in LP for arguing in AF. These frameworks are extended to *bidirectional frameworks* in which AF and LP can exchange information with each other. We also investigate their connection to several general KR frameworks from the literature.

1 Introduction

A *logic program* (LP) represents declarative knowledge as a set of rules and realizes commonsense reasoning as logical inference. An *argumentation framework* (AF), on the other hand, represents arguments and an attack relation over them, and defines acceptable arguments under various semantics. The two frameworks specify different types of knowledge and realize different types of reasoning. In our daily life, however, we often use two modes of reasoning interchangeably. For instance, consider a logic program $LP = \{\, get\_vaccine \leftarrow safe \wedge effective,\ \ \neg get\_vaccine \leftarrow not\ safe \,\}$ which says that we get a vaccine if it is safe and effective, and we do not get it if it is not safe. To see whether a vaccine is safe and effective, we refer to an expert opinion. It is often the case, however, that multiple experts have different opinions. In this case, we observe argumentation among experts and take it into account to make a decision. In other words, the truth value of *safe* is determined by an external argumentation framework such as $AF = (\{s,d\}, \{(s,d),(d,s)\})$ in its most condensed form where s represents safe and d represents dangerous. A credulous reasoner will accept *safe* under the stable semantics, while a skeptical reasoner will not accept it under the grounded semantics. A reasoner determines acceptable arguments under chosen semantics and makes a decision using his/her own LP. For another example, consider a debate on whether global warming is occurring. Scientists and politicians make different claims based on evidence and scientific knowledge. An argumentation framework is used for representing the debate, while

The second author has been partial supported by NSF grants 1914635, 1757207, 1812628.

G. Gottlob et al. (Eds.): LPNMR 2022, LNAI 13416, pp. 322–335, 2022.
https://doi.org/10.1007/978-3-031-15707-3_25

arguments appearing in the argumentation graph are generated as results of reasoning from the background knowledge of participants represented by LPs.

In these examples, we can encode reasoners' private knowledge as LPs and argumentation in the public space as AFs. It is natural to distinguish two different types of knowledge and interlink them with each other. In the first example, an agent has a private knowledge base that refers to opinions in a public argumentation framework. In the second example, on the other hand, agents participating in a debate have their private knowledge bases supporting their individual claims.

Logic programs and argumentation frameworks are mutually transformed with each other. Dung [6] provides a transformation from LPs to AFs and shows that stable models [11] (resp. the well-founded model [16]) of a logic program correspond to stable extensions (resp. the grounded extension) of a transformed argumentation framework. He also introduces a converse transformation from AFs to LPs, and shows that the semantic correspondences still hold. The results are extended to equivalences of LPs and AFs under different semantics (e.g. [5]). Using such transformational approaches, an LP and an AF can be combined and one could perform both argumentative reasoning and commonsense reasoning in a single framework. One of the limitations of this approach is that in order to combine an LP and an AF into a single framework, the two frameworks must have the corresponding semantics. For instance, suppose that an agent has a knowledge base *LP* and refers to an *AF*. If the agent uses the stable model semantics of *LP*, then to combine *LP* with *AF* using a transformation proposed in [5,6] *AF* must use the stable extension semantics. Argumentation can have an internal structure in *structured argumentation*. In *assumption based argumentation* (ABA) [7], for instance, an argument for a claim c is supported by a set of assumptions S if c is deduced from S using a set of LP rules ($S \vdash c$). A structured argumentation has a knowledge base inside an argument and provides reasons that support particular claims. An argument is represented as a tree and an attack relation is introduced between trees. However, merging argumentation and knowledge bases into a single framework would produce a huge argumentation structure that is complicated and hard to manage.

In this paper, we introduce new frameworks, called *LPAF* and *AFLP*, for interlinking LPs and AFs. The *LPAF* uses the result of argumentation in AFs for reasoning in LPs. In contrast, the *AFLP* uses the result of reasoning in LPs for arguing in AFs. These frameworks are extended to *bidirectional frameworks* in which AFs and LPs can exchange information with each other. We address applications of the proposed framework and investigate connections to existing KR frameworks. The rest of this paper is organized as follows. Section 2 reviews basic notions of logic programming and argumentation frameworks. Section 3 introduces several frameworks for interlinking LPs and AFs. Section 4 presents applications to several KR frameworks. Section 5 discusses complexity issues and Sect. 6 summarizes the paper. Due to space limitation, proofs of propositions are omitted in this paper. They are available in the longer version [15].

2 Preliminaries

We consider a language that contains a finite set $\mathscr{L}$ of propositional variables.

Definition 1. A *(disjunctive) logic program* (LP) is a finite set of *rules* of the form:

$$p_1 \vee \cdots \vee p_\ell \leftarrow q_1, \ldots, q_m, not\, q_{m+1}, \ldots, not\, q_n \quad (\ell, m, n \geq 0)$$

where p_i and q_j are propositional variables in $\mathscr{L}$ and *not* is *negation as failure* (NAF).

The left-hand side of $\leftarrow$ is the *head* and the right-hand side is the *body*. For each rule r of the above form, $head(r)$, $body^+(r)$, and $body^-(r)$ respectively denote the sets of atoms $\{p_1, \ldots, p_\ell\}$, $\{q_1, \ldots, q_m\}$, and $\{q_{m+1}, \ldots, q_n\}$, and $body(r) = body^+(r) \cup body^-(r)$. A *(disjunctive) fact* is a rule r with $body(r) = \emptyset$. A fact is a *non-disjunctive fact* if $\ell = 1$. An LP is a *normal logic program* if $|\, head(r)\, | \leq 1$ for any rule r in the program. Given a logic program LP, put $Head(LP) = \bigcup_{r \in LP} head(r)$ and $Body(LP) = \bigcup_{r \in LP} body(r)$. Throughout the paper, a program means a propositional/ground logic program and $\mathscr{B}_{LP}$ is the set of ground atoms appearing in a program LP (called the *Herbrand base*).

A program LP under the μ semantics is denoted by LP_μ. The semantics of LP_μ is defined as the set $\mathscr{M}^\mu_{LP} \subseteq 2^{\mathscr{B}_{LP}}$ (or simply $\mathscr{M}^\mu$) of μ models of LP. If a ground atom p is included in every μ model of LP, we write $LP_\mu \models p$. LP_μ is simply written as LP if the semantics is clear in the context. A logic programming semantics μ is *universal* if every LP has a μ model. The stable model semantics is not universal, while the *well-founded semantics* of normal logic programs is universal.A logic program LP under the stable model semantics (resp. well-founded semantics) is written as LP_{stb} (resp. LP_{wf}).

Definition 2. An *argumentation framework* (AF) is a pair (A, R) where $A \subseteq \mathscr{L}$ is a finite set of *arguments* and $R \subseteq A \times A$ is an *attack relation*.

For an AF (A, R), we say that an argument *a attacks* an argument b if $(a, b) \in R$. A set S of arguments *attacks* an argument a iff there is an argument $b \in S$ that attacks a; S is *conflict-free* if there are no arguments $a, b \in S$ such that a attacks b. S *defends* an argument a if S attacks every argument that attacks a. We write $D(S) = \{\, a \mid S \text{ defends } a \,\}$.

The semantics of AF is defined as the set of designated *extensions* [6]. Given $AF = (A, R)$, a conflict-free set of arguments $S \subseteq A$ is a *complete extension* iff $S = D(S)$; a *stable extension* iff S attacks each argument in $A \setminus S$; a *preferred extension* iff S is a maximal complete extension of AF (wrt $\subseteq$); a *grounded extension* iff S is the minimal complete extension of AF (wrt $\subseteq$). An argumentation framework AF under the ω semantics is denoted by AF_ω. The semantics of AF_ω is defined as the set $\mathscr{E}^\omega_{AF}$ (or simply $\mathscr{E}^\omega$) of ω extensions of AF. We abbreviate the above four semantics of AF as AF_{com}, AF_{stb}, AF_{prf} and AF_{grd}, respectively. AF_ω is simply written as AF if the semantics is clear in the context. Among the four semantics, the following relations hold: for any AF, $\mathscr{E}^{stb}_{AF} \subseteq \mathscr{E}^{prf}_{AF} \subseteq \mathscr{E}^{com}_{AF}$ and $\mathscr{E}^{grd}_{AF} \subseteq \mathscr{E}^{com}_{AF}$. $\mathscr{E}^{stb}_{AF}$ is possibly empty, while others are not. In particular, $\mathscr{E}^{grd}_{AF}$ is a singleton set. An argumentation semantics ω is *universal* if every AF has an ω extension. The stable semantics is not universal, while the other three semantics presented above are universal. [1]

[1] We assume readers familiarity with the stable model semantics [11], [14] and the well-founded semantics [16]

3 Linking LP and AF

3.1 From AF to LP

We first introduce a framework that can use the result of argumentation in AFs for reasoning in LPs. In this subsection, we assume that $Head(LP) \cap A = \varnothing$ for a program LP and $AF = (A, R)$, that is, no rule in a logic program has an argument in its head.

Definition 3. Given an LP and $AF = (A, R)$, define $LP^{+A} = \{ r \in LP \mid body(r) \cap A \neq \varnothing \}$ and $LP^{-A} = \{ r \in LP \mid body(r) \cap A = \varnothing \}$. We say that each rule in LP^{+A} (resp. LP^{-A}) *refers to* arguments (resp. is *free from* arguments). An argument $a \in A$ is *referred to* in LP if a appears in LP. Define $A|_{LP} = \{ a \in A \mid a$ is referred to in $LP \}$.

By definition, an LP is partitioned into $LP = LP^{+A} \cup LP^{-A}$.

Definition 4. Given an LP and $AF = (A, R)$, a μ *model of LP extended by* $\mathscr{A} \subseteq 2^A$ is a μ model of $LP \cup \{ a \leftarrow \mid a \in E \cap A|_{LP} \}$ for some $E \in \mathscr{A}$ if $\mathscr{A} \neq \varnothing$; otherwise, it is a μ model of LP^{-A}.

Definition 5. A *simple LPAF framework* is a pair $\langle LP_\mu, AF_\omega \rangle$, where LP_μ is a program under the μ semantics and AF_ω is an argumentation framework under the ω semantics.

Definition 6. Let $\varphi = \langle LP_\mu, AF_\omega \rangle$ be a simple LPAF framework. Suppose that AF has the set of ω extensions: $\mathscr{E}^\omega = \{E_1, \ldots, E_k\}$ $(k \geq 0)$. Then an *LPAF model* of φ is defined as a μ model of LP_μ extended by $\mathscr{E}^\omega$. The set of LPAF models of φ is denoted by $\mathbf{M}_\varphi$.

By definition, an LPAF model is defined as a μ model of the program LP by introducing arguments that are referred to in LP and are acceptable under the ω semantics of AF. If the AF part has no ω extension ($\mathscr{E}^\omega = \varnothing$), on the other hand, AF provides no justification for arguments referred to by LP. In this case, we do not take the consequences that are derived using arguments in AF. Then an LPAF model is constructed by rules that are free from arguments in AF.

Example 1. Consider $\varphi_1 = \langle LP_{stb},\ AF_{stb} \rangle$ where $LP_{stb} = \{ p \leftarrow a,\quad q \leftarrow not\, a \}$ and $AF_{stb} = (\{a, b\}, \{(a, b), (b, a)\})$. As AF_{stb} has two stable extensions $\{a\}$ and $\{b\}$, φ_1 has two LPAF models $\{p, a\}$ and $\{q\}$. On the other hand, if we use $\omega = grounded$ then AF_{grd} has the single extension $\varnothing$. Then $\langle LP_{stb},\ AF_{grd} \rangle$ has the single LPAF model $\{q\}$.[2] Next, consider $\varphi_2 = \langle LP_{stb},\ AF_{stb} \rangle$ where $LP_{stb} = \{ p \leftarrow not\, a,\quad q \leftarrow not\, p \}$ and $AF_{stb} = (\{a, b\}, \{(a, b), (a, a)\})$. As AF_{stb} has no stable extension and the second rule in LP_{stb} is free from arguments, φ_2 has the single LPAF model $\{q\}$. Note that if we keep the first rule then a different conclusion p is obtained from LP_{stb}. We do not consider the conclusion justified because AF_{stb} provides no information on whether the argument a is acceptable or not.

Proposition 1. *Let $\varphi_1 = \langle LP_\mu, AF^1_{\omega_1} \rangle$ and $\varphi_2 = \langle LP_\mu, AF^2_{\omega_2} \rangle$ be two LPAFs such that $\mathscr{E}^{\omega_1}_{AF^1} \neq \varnothing$. If $\mathscr{E}^{\omega_1}_{AF^1} \subseteq \mathscr{E}^{\omega_2}_{AF^2}$, then $\mathbf{M}_{\varphi_1} \subseteq \mathbf{M}_{\varphi_2}$.*

[2] Note that an AF extension represents whether an argument is accepted or not. If an argument a is not in an extension E, a is not accepted in E. Then $not\, a$ in LP becomes true by NAF.

Proposition 1 implies the inclusion relations with the same AF under different semantics: $\mathbf{M}_{\varphi_1} \subseteq \mathbf{M}_{\varphi_2}$ holds for $\varphi_1 = \langle LP_\mu, AF_{prf} \rangle$ and $\varphi_2 = \langle LP_\mu, AF_{com} \rangle$; $\varphi_1 = \langle LP_\mu, AF_{stb} \rangle$ and $\varphi_2 = \langle LP_\mu, AF_{prf} \rangle$; or $\varphi_1 = \langle LP_\mu, AF_{grd} \rangle$ and $\varphi_2 = \langle LP_\mu, AF_{com} \rangle$.

Two programs LP^1_μ and LP^2_μ are *uniformly equivalent relative to* A (denoted $LP^1_\mu \equiv^A_u LP^2_\mu$) if for any set of non-disjunctive facts $F \subseteq A$, the programs $LP^1_\mu \cup F$ and $LP^2_\mu \cup F$ have the same set of μ models [10]. The equivalence of two simple LPAF frameworks is then characterized as follows.

Proposition 2. *Let $\varphi_1 = \langle LP^1_\mu, AF_\omega \rangle$ and $\varphi_2 = \langle LP^2_\mu, AF_\omega \rangle$ be two LPAFs such that $\mathscr{E}^\omega \neq \varnothing$. Then, $\mathbf{M}_{\varphi_1} = \mathbf{M}_{\varphi_2}$ if $LP^1_\mu \equiv^A_u LP^2_\mu$ and $A|_{LP^1_\mu} = A|_{LP^2_\mu}$ where $AF_\omega = (A, R)$.*

A simple LPAF framework $\varphi = \langle LP_\mu, AF_\omega \rangle$ is *consistent* if φ has an LPAF model. The consistency of φ depends on the chosen semantics μ. In particular, a simple LPAF framework $\varphi = \langle LP_\mu, AF_\omega \rangle$ is consistent if μ is universal. $\varphi = \langle LP_\mu, AF_\omega \rangle$ may have an LPAF model even if $\mathscr{M}^\mu_{LP} = \mathscr{E}^\omega_{AF} = \varnothing$.

Example 2. Consider $\varphi = \langle LP_{stb}, AF_{stb} \rangle$ where $LP_{stb} = \{\, p \leftarrow not\, a, not\, p, \quad q \leftarrow \,\}$ and $AF_{stb} = (\{a\}, \{(a,a)\})$. Then $\mathscr{M}^{stb}_{LP} = \mathscr{E}^{stb}_{AF} = \varnothing$, but φ has the LPAF model $\{q\}$.

A simple LPAF consists of a single LP and an AF, which is generalized to a framework that consists of multiple LPs and AFs.

Definition 7. A *general LPAF framework* is defined as a tuple $\langle \mathbb{LP}^m, \mathbb{AF}^n \rangle$ where $\mathbb{LP}^m = (LP^1_{\mu_1}, \ldots, LP^m_{\mu_m})$ and $\mathbb{AF}^n = (AF^1_{\omega_1}, \ldots, AF^n_{\omega_n})$. Each $LP^i_{\mu_i}$ $(1 \leq i \leq m)$ is a logic program LP^i under the μ_i semantics and each $AF^j_{\omega_j}$ $(1 \leq j \leq n)$ is an argumentation framework AF^j under the ω_j semantics.

A general LPAF framework is used in a situation where multiple agents have individual LPs as their private knowledge bases and each agent possibly refers to the results of argumentation of open AFs. The semantics of a general LPAF is defined as an extension of a simple LPAF framework.

Definition 8. Let $\varphi = \langle \mathbb{LP}^m, \mathbb{AF}^n \rangle$ be a general LPAF framework. The *LPAF state* of φ is defined as a tuple $(\Sigma_1, \ldots, \Sigma_m)$ where $\Sigma_i = (\mathbf{M}^i_1, \ldots, \mathbf{M}^i_n)$ $(1 \leq i \leq m)$ and $\mathbf{M}^i_j$ $(1 \leq j \leq n)$ is the set of LPAF models of $\langle LP^i_{\mu_i}, AF^j_{\omega_j} \rangle$.

By definition, an LPAF state consists of a collection of LPAF models such that each model is obtained by combining a program $LP^i_{\mu_i}$ and an argumentation framework $AF^j_{\omega_j}$.

Example 3. Consider $\varphi = \langle (LP_{stb}, LP_{wf}), (AF_{stb}, AF_{grd}) \rangle$ where $LP_{stb} = LP_{wf} = \{\, p \leftarrow a, not\, q, \quad q \leftarrow a, not\, p \,\}$ and $AF_{stb} = AF_{grd} = (\{a,b\}, \{(a,b),(b,a)\})$. In this case, $\langle LP_{stb}, AF_{stb} \rangle$ has three LPAF models: $\{p,a\}$, $\{q,a\}$ and $\varnothing$; $\langle LP_{stb}, AF_{grd} \rangle$ has the single LPAF model: $\varnothing$; $\langle LP_{wf}, AF_{stb} \rangle$ has two LPAF models:[3] $\{a\}$ and $\varnothing$; $\langle LP_{wf}, AF_{grd} \rangle$ has the single LPAF model: $\varnothing$. Then φ has the LPAF state (Σ_1, Σ_2) where $\Sigma_1 = (\{\{p,a\}, \{q,a\}, \varnothing\}, \{\varnothing\})$ and $\Sigma_2 = (\{\{a\}, \varnothing\}, \{\varnothing\})$.

[3] We consider the well-founded model as the set of true atoms under the well-founded semantics.

The above example shows that a general LPAF is used for comparing the results of combination between LP and AF under different semantics. Given tuples $(S_1,\ldots,S_k)$ and $(T_1,\ldots,T_\ell)$ $(k,\ell \geq 1)$, define $(S_1,\ldots,S_k) \oplus (T_1,\ldots,T_\ell) = (S_1,\ldots,S_k,T_1,\ldots,T_\ell)$.

Proposition 3. *Let $\varphi = \langle \mathbb{LP}^m, \mathbb{AF}^n \rangle$ be a general LPAF framework. Then the LPAF state $(\Sigma_1,\ldots,\Sigma_m)$ of φ is obtained by $(\Sigma_1,\ldots,\Sigma_k) \oplus (\Sigma_{k+1},\ldots,\Sigma_m)$ $(1 \leq k \leq m-1)$ where $(\Sigma_1,\ldots,\Sigma_k)$ is the LPAF state of $\varphi_1 = \langle \mathbb{LP}^k,\ \mathbb{AF}^n \rangle$ and $(\Sigma_{k+1},\ldots,\Sigma_m)$ is the LPAF state of $\varphi_2 = \langle \mathbb{LP}^m_{k+1},\ \mathbb{AF}^n \rangle$ where $\mathbb{LP}^m_{k+1} = (LP^{k+1}_{\mu_{k+1}},\ldots,LP^m_{\mu_m})$.*

Proposition 3 presents that a general LPAF has the modularity property; φ is partitioned into smaller φ_1 and φ_2, and the introduction of new LPs to φ is done incrementally.

3.2 From LP to AF

We next introduce a framework that can use the result of reasoning in LPs for arguing in AFs. In this subsection, we assume that $Body(LP) \cap A = \varnothing$ for a program LP and $AF = (A,R)$, that is, no rule in a logic program has an argument in its body.

Definition 9. Let $AF = (A,R)$ and $M \subseteq \mathscr{L}$. Then AF *with support* M is defined as $AF^M = (A,R')$ where $R' = R \setminus \{\,(x,a) \mid x \in A \text{ and } a \in A \cap M\,\}$.

By definition, AF^M is an argumentation framework in which every tuple attacking $a \in M$ is removed from R. As a result, every argument included in M is accepted in AF^M.

Definition 10. Let $AF = (A,R)$ and $\mathscr{M} \subseteq 2^{\mathscr{B}_{LP}}$. An *$\omega$ extension of AF supported by $\mathscr{M}$* is an ω extension of AF^M for some $M \in \mathscr{M}$ if $\mathscr{M} \neq \varnothing$; otherwise, it is an ω extension of (A',R') where $A' = A \setminus \mathscr{B}_{LP}$ and $R' = R \cap (A' \times A')$.

Definition 11. A *simple AFLP framework* is a pair $\langle AF_\omega, LP_\mu \rangle$ where AF_ω is an argumentation framework under the ω semantics and LP_μ is a program under μ semantics.

Definition 12. Let $\psi = \langle AF_\omega, LP_\mu \rangle$ be a simple AFLP framework and $\mathscr{M}^\mu \subseteq 2^{\mathscr{B}_{LP}}$ be the set of μ models of LP. An *AFLP extension* of ψ is defined as an ω extension of AF_ω supported by $\mathscr{M}^\mu$. $\mathbf{E}_\psi$ denotes the set of AFLP extensions of ψ.

By definition, an AFLP extension is defined as an ω extension of AF^M_ω that takes into account support information in a μ model M of LP. If the LP part has no μ model ($\mathscr{M}^\mu = \varnothing$), on the other hand, LP provides no ground for arguments in $A \cap \mathscr{B}_{LP}$. In this case, we do not use those arguments that rely on LP. Then an AFLP extension is constructed using arguments that do not appear in LP.

Example 4. Consider $\psi_1 = \langle AF_{stb}, LP_{stb} \rangle$ where $AF_{stb} = (\{a,b\},\{(a,b),(b,a)\})$ and $LP_{stb} = \{\,a \leftarrow p,\ \ p \leftarrow not\, q,\ \ q \leftarrow not\, p\,\}$. LP_{stb} has two stable models $M_1 = \{a,p\}$ and $M_2 = \{q\}$, then $AF^{M_1}_{stb} = (\{a,b\},\{(a,b)\})$ and $AF^{M_2}_{stb} = AF_{stb}$. Hence, ψ_1 has two AFLP extensions $\{a\}$ and $\{b\}$. On the other hand, if we use $\omega = grounded$, then $\langle AF_{grd}, LP_{stb} \rangle$ has two AFLP extensions $\{a\}$ and $\varnothing$. Next, consider $\psi_2 = \langle AF_{grd}, LP_{stb} \rangle$ where $AF_{grd} = (\{a,b,c\},\{(a,b),(b,c)\})$ and $LP_{stb} = \{\,a \leftarrow p,\quad p \leftarrow not\, p\,\}$. As LP_{stb} has no stable model, ψ_2 has the AFLP extension $\{b\}$ as the grounded extension of $(\{b,c\},\{(b,c)\})$.

Proposition 4. *Let* $\psi_1 = \langle AF_\omega, LP^1_{\mu_1} \rangle$ *and* $\psi_2 = \langle AF_\omega, LP^2_{\mu_2} \rangle$ *be two AFLPs such that* $\mathscr{M}^{\mu_1}_{LP^1} \neq \emptyset$. *If* $\mathscr{M}^{\mu_1}_{LP^1} \subseteq \mathscr{M}^{\mu_2}_{LP^2}$, *then* $\mathbf{E}_{\psi_1} \subseteq \mathbf{E}_{\psi_2}$.

Baumann [1] introduces equivalence relations of AFs with respect to deletion of arguments and attacks. For two $AF^1_\omega = (A_1, R_1)$ and $AF^2_\omega = (A_2, R_2)$, AF^1_ω and AF^2_ω are *normal deletion equivalent* (denoted by $AF^1_\omega \equiv_{nd} AF^2_\omega$) if for any set A of arguments $(A'_1, R_1 \cap (A'_1 \times A'_1))$ and $(A'_2, R_2 \cap (A'_2 \times A'_2))$ have the same set of ω extensions where $A'_1 = A_1 \setminus A$ and $A'_2 = A_2 \setminus A$. In contrast, AF^1_ω and AF^2_ω are *local deletion equivalent* (denoted by $AF^1_\omega \equiv_{ld} AF^2_\omega$) if for any set R of attacks $(A_1, R_1 \setminus R)$ and $(A_2, R_2 \setminus R)$ have the same set of ω extensions. By definition, we have the next result.

Proposition 5. *Let* $\psi_1 = \langle AF^1_\omega, LP_\mu \rangle$ *and* $\psi_2 = \langle AF^2_\omega, LP_\mu \rangle$ *be two AFLPs. Then,* $\mathbf{E}_{\psi_1} = \mathbf{E}_{\psi_2}$ *if (i)* $\mathscr{M}^\mu = \emptyset$ *and* $AF^1_\omega \equiv_{nd} AF^2_\omega$; *or (ii)* $\mathscr{M}^\mu \neq \emptyset$ *and* $AF^1_\omega \equiv_{ld} AF^2_\omega$.

Baumann shows that $AF^1_\omega \equiv_{ld} AF^2_\omega$ if and only if $AF^1_\omega = AF^2_\omega$ for any $\omega = \{com, stb, prf, grd\}$. In contrast, necessary or sufficient conditions for $AF^1_\omega \equiv_{nd} AF^2_\omega$ are given by the structure of argumentation graphs and they differ from the chosen semantics in general.

A simple AFLP framework $\psi = \langle AF_\omega, LP_\mu \rangle$ is *consistent* if ψ has an AFLP extension. By definition, a simple AFLP framework $\psi = \langle AF_\omega, LP_\mu \rangle$ is consistent if ω is universal. A simple AFLP consists of a single AF and an LP, which is generalized to a framework that consists of multiple AFs and LPs.

Definition 13. A *general AFLP framework* is defined as a tuple $\langle \mathbb{AF}^n, \mathbb{LP}^m \rangle$ where $\mathbb{AF}^n = (AF^1_{\omega_1}, \ldots, AF^n_{\omega_n})$ and $\mathbb{LP}^m = (LP^1_{\mu_1}, \ldots, LP^m_{\mu_m})$. Each $AF^j_{\omega_j}$ $(1 \leq j \leq n)$ is an argumentation framework AF^j under the ω_j semantics and each $LP^i_{\mu_i}$ $(1 \leq i \leq m)$ is a logic program LP^i under the μ_i semantics.

A general AFLP framework is used in a situation such that argumentative dialogues consult LPs as information sources. The semantics of a general AFLP is defined as an extension of a simple AFLP framework.

Definition 14. Let $\psi = \langle \mathbb{AF}^n, \mathbb{LP}^m \rangle$ be a general AFLP framework. The *AFLP state* of ψ is defined as a tuple $(\Gamma_1, \ldots, \Gamma_n)$ where $\Gamma_j = (\mathbf{E}^j_1, \ldots, \mathbf{E}^j_m)$ $(1 \leq j \leq n)$ and $\mathbf{E}^j_i$ $(1 \leq i \leq m)$ is the set of AFLP extensions of $\langle AF^j_{\omega_j}, LP^i_{\mu_i} \rangle$.

By definition, an AFLP state consists of a collection of AFLP extensions such that each extension is obtained by combining $AF^j_{\omega_j}$ and $LP^i_{\mu_i}$.

Example 5. Consider $\psi = \langle (AF_{grd}), (LP^1_{stb}, LP^2_{stb}) \rangle$ where $AF_{grd} = (\{a, b\}, \{(a, b)\})$, $LP^1_{stb} = \{a \leftarrow p, \quad p \leftarrow\}$, and $LP^2_{stb} = \{b \leftarrow q, \quad q \leftarrow\}$. Then, $\langle AF_{grd}, LP^1_{stb} \rangle$ has the AFLP extension $\{a\}$, while $\langle AF_{grd}, LP^2_{stb} \rangle$ has the AFLP extension $\{a, b\}$. Then the AFLP state of ψ is (Γ_1) where $\Gamma_1 = (\{\{a\}\}, \{\{a, b\}\})$.

A general AFLP has the modularity property. The operation $\oplus$ is defined in Sect. 3.1.

Proposition 6. *Let* $\psi = \langle \mathbb{AF}^n, \mathbb{LP}^m \rangle$ *be a general AFLP framework. Then the AFLP state* $(\Gamma_1, \ldots, \Gamma_n)$ *of* ψ *is obtained by* $(\Gamma_1, \ldots, \Gamma_k) \oplus (\Gamma_{k+1}, \ldots, \Gamma_n)$ $(1 \leq k \leq n-1)$ *where* $(\Gamma_1, \ldots, \Gamma_k)$ *is the AFLP state of* $\psi_1 = \langle \mathbb{AF}^k, \mathbb{LP}^m \rangle$ *and* $(\Gamma_{k+1}, \ldots, \Gamma_n)$ *is the AFLP state of* $\psi_2 = \langle \mathbb{AF}^n_{k+1}, \mathbb{LP}^m \rangle$ *where* $\mathbb{AF}^n_{k+1} = (AF^{k+1}_{\omega_{k+1}}, \ldots, AF^n_{\omega_n})$.

3.3 Bidirectional Framework

In Sects. 3.1 and 3.2 we provided frameworks in which given LPs and AFs one refers the other in one direction. This subsection provides a framework such that LPs and AFs interact with each other. Such a situation happens in social media, for instance, where a person posts his/her opinion to an Internet forum, which arises public discussion on the topic, then the person revises his/her belief by the result of discussion. In this subsection, we assume that any rule in LP could contain arguments in its head or body.

Definition 15. A *simple bidirectional LPAF framework* is defined as a pair $\langle\langle LP_\mu, AF_\omega \rangle\rangle$ where LP_μ is a logic program and AF_ω is an argumentation framework.

Definition 16. Let $\zeta = \langle\langle LP_\mu, AF_\omega \rangle\rangle$ be a simple bidirectional LPAF framework. Suppose that a simple AFLP framework $\psi = \langle AF_\omega, LP_\mu \rangle$ has the set of AFLP extensions $\mathbf{E}_\psi$. Then a *BDLPAF model* of ζ is defined as a μ model of LP_μ extended by $\mathbf{E}_\psi$.

BDLPAF models reduce to LPAF models if $\mathbf{E}_\psi$ coincides with $\mathscr{E}^\omega_{AF}$. In the bidirectional framework, an LP can refer to arguments in AF and AF can get a support from the LP.

Example 6. Consider $\zeta = \langle\langle LP_{stb},\ AF_{stb} \rangle\rangle$ where $LP_{stb} = \{\, a \leftarrow not\, p, \quad q \leftarrow c \,\}$ and $AF_{stb} = (\{a,b,c\}, \{(a,b),(b,a),(b,c)\})$. The simple AFLP framework $\langle AF_{stb}, LP_{stb} \rangle$ has the AFLP extension $E = \{a,c\}$. So, the BDLPAF model of ζ becomes $\{a,c,q\}$.

Similarly, we can make a simple AFLP bidirectional.

Definition 17. A *simple bidirectional AFLP framework* is defined as a pair $\langle\langle AF_\omega, LP_\mu \rangle\rangle$ where AF_ω is an argumentation framework and LP_μ is a logic program.

Definition 18. Let $\eta = \langle\langle AF_\omega, LP_\mu \rangle\rangle$ be a simple bidirectional AFLP framework. Suppose that a simple LPAF framework $\varphi = \langle LP_\mu, AF_\omega \rangle$ has the set of LPAF models $\mathbf{M}_\varphi$. Then a *BDAFLP extension* of η is defined as an ω extension of AF_ω supported by $\mathbf{M}_\varphi$.

Example 7. Consider $\eta = \langle\langle AF_{grd},\ LP_{stb} \rangle\rangle$ where $AF_{grd} = (\{a,b\}, \{(a,b),(b,a)\})$ and $LP_{stb} = \{\, p \leftarrow a, \quad q \leftarrow not\, a, \quad b \leftarrow q \,\}$. The simple LPAF framework $\langle LP_{stb}, AF_{grd} \rangle$ has the single LPAF model $M = \{b,q\}$. So, the BDAFLP extension of η becomes $\{b\}$.

Given AF_ω and LP_μ, a series of BDLPAF models (or BDAFLP extensions) can be built by repeatedly referring to each other. Starting with the AFLP extensions $\mathbf{E}^0_\psi$, the BDLPAF models $\mathbf{M}^1_\varphi$ extended by $\mathbf{E}^0_\psi$ are produced, then the BDAFLP extensions $\mathbf{E}^1_\psi$ supported by $\mathbf{M}^1_\varphi$ are produced, which in turn produce the BDLPAF models $\mathbf{M}^2_\varphi$ extended by $\mathbf{E}^1_\psi$, and so on. Likewise, starting with the LPAF models $\mathbf{M}^0_\varphi$, the sets $\mathbf{E}^1_\psi$, $\mathbf{M}^1_\varphi$, $\mathbf{E}^2_\psi$, ..., are produced. We write the sequences of BDLPAF models and BDAFLP extensions as $[\mathbf{M}^1_\varphi, \mathbf{M}^2_\varphi, \ldots]$ and $[\mathbf{E}^1_\psi, \mathbf{E}^2_\psi, \ldots]$, respectively.

Proposition 7. *Let* $[\mathbf{M}^1_\varphi, \mathbf{M}^2_\varphi, \ldots]$ *and* $[\mathbf{E}^1_\psi, \mathbf{E}^2_\psi, \ldots]$ *be sequences defined as above. Then,* $\mathbf{M}^i_\varphi = \mathbf{M}^{i+1}_\varphi$ *and* $\mathbf{E}^j_\psi = \mathbf{E}^{j+1}_\psi$ *for some* $i, j \geq 1$.

4 Applications

4.1 Deductive Argumentation

A *structured argumentation* is a framework such that there is an internal structure to an argument. In structured argumentation, knowledge is represented using a formal language and each argument is constructed from that knowledge. Given a logical language $\mathscr{L}$ and a consequence relation $\vdash$ in $\mathscr{L}$, a *deductive argument* [2] is a pair $\langle \mathscr{F}, c \rangle$ where $\mathscr{F}$ is a set of formulas in $\mathscr{L}$ and c is a (ground) atom such that $\mathscr{F} \vdash c$. $\mathscr{F}$ is called the *support* of the argument and c is the *claim*. A *counterargument* is an argument that attacks another argument. It is defined in terms of logical contradiction between the claim of a counterargument and the premises of the claim of an attacked argument.

An AFLP framework is captured as a kind of deductive arguments in the sense that LP can support an argument a appearing in AF. There is an important difference, however. In an AFLP, argumentative reasoning in AF and deductive reasoning in LP are separated. The AF part is kept at the abstract level and the LP part represents reasons for supporting particular arguments. As such, an AFLP provides a middle ground between abstract argumentation and structured argumentation. Such a separation keeps the whole structure compact and makes it easy to update AF or LP without changing the other part. Thus, AFLP/LPAF supports an elaboration tolerant development of knowledge bases. This allows us to characterize deductive argumentation in AFLP as follows.

Definition 19. Let $\psi = \langle \mathbb{AF}^n, \mathbb{LP}^m \rangle$ be a general AFLP framework s.t. $AF^i_{\omega_i} = (A^i, R^i)$ $(1 \leq i \leq n)$. (i) $a \in A^i$ is *supported* in $LP^j_{\mu_j}$ for some $1 \leq j \leq m$ (written $(LP^j_{\mu_j}, a)$) if $LP^j_{\mu_j} \models a$; (ii) $(LP^j_{\mu_j}, a)$ and $(LP^k_{\mu_k}, b)$ *rebut* each other if $\{(a,b), (b,a)\} \subseteq R^i$ for some i; (iii) $(LP^j_{\mu_j}, a)$ *undercuts* $(LP^k_{\mu_k}, b)$ if $LP^k_{\mu_k} \cup \{a\} \not\models b$.

Example 8. ([2]) (a) There is an argument that the government should cut spending because of a budget deficit. On the other hand, there is a counterargument that the government should not cut spending because the economy is weak. These arguments are respectively represented using deductive arguments as: $A1 = \langle \{\mathit{deficit},\ \mathit{deficit} \rightarrow \mathit{cut}\}, \mathit{cut} \rangle$ and $A2 = \langle \{\mathit{weak},\ \mathit{weak} \rightarrow \neg \mathit{cut}\}, \neg \mathit{cut} \rangle$ where $A1$ and $A2$ rebut each other. The situation is represented using the AFLP $\langle (AF_{stb}), (LP^1_{stb}, LP^2_{stb}) \rangle$ such that $AF_{stb} = (\{\mathit{cut}, \mathit{no\text{-}cut}\},\ \{(\mathit{cut}, \mathit{no\text{-}cut}),\ (\mathit{no\text{-}cut}, \mathit{cut})\})$; $LP^1_{stb} = \{\mathit{cut} \leftarrow \mathit{deficit},\quad \mathit{deficit} \leftarrow\}$; $LP^2_{stb} = \{\mathit{no\text{-}cut} \leftarrow \mathit{weak},\quad \mathit{weak} \leftarrow\}$. Then $(LP^1_{stb}, \mathit{cut})$ and $(LP^2_{stb}, \mathit{no\text{-}cut})$ rebut each other.

(b) There is an argument that the metro is an efficient (*eff*) form of transport, so one can use it. On the other hand, there is a counterargument that the metro is inefficient (*ineff*) because of a strike. These arguments are respectively represented using deductive arguments as: $A1 = \langle \{\mathit{eff},\ \mathit{eff} \rightarrow \mathit{use}\}, \mathit{use} \rangle$ and $A2 = \langle \{\mathit{strike},\ \mathit{strike} \rightarrow \neg \mathit{eff}\}, \neg \mathit{eff} \rangle$ where $A2$ undercuts $A1$. The situation is represented using an AFLP $\langle (AF_{stb}), (LP^1_{stb}, LP^2_{stb}) \rangle$ such that $AF_{stb} = (\{\mathit{eff}, \mathit{ineff}\},\ \{(\mathit{eff}, \mathit{ineff}), (\mathit{ineff}, \mathit{eff})\})$; $LP^1_{stb} = \{\mathit{use} \leftarrow \mathit{eff},\quad \mathit{eff} \leftarrow \mathit{not}\ \mathit{ineff}\}$; $LP^2_{stb} = \{\mathit{ineff} \leftarrow \mathit{strike},\quad \mathit{strike} \leftarrow\}$. Then $(LP^2_{stb}, \mathit{ineff})$ undercuts $(LP^1_{stb}, \mathit{use})$.

4.2 Argument Aggregation

Argument aggregation or *collective argumentation* [3] considers a situation in which multiple agents may have different arguments and/or opinions. The problems are then what and how to aggregate arguments. In abstract argumentation, the problem is formulated as follows. Given several AFs having different arguments and attacks, find acceptable arguments among those AFs. In the *argument-wise aggregation*, individually supported arguments are aggregated by some voting mechanism.

Example 9. ([3]) Suppose three agents deciding which among three arguments a, b, and c, are collectively acceptable. Each agent has a subjective evaluation of the interaction among those arguments, leading to three different individual AFs: $AF_1 = (\{a,b,c\}, \{(a,b),(b,c)\})$, $AF_2 = (\{a,b,c\}, \{(a,b)\})$, and $AF_3 = (\{a,b,c\}, \{(b,c)\})$. Three AFs have the grounded extensions $\{a,c\}$, $\{a,c\}$, and $\{a,b\}$, respectively. By majority voting, $\{a,c\}$ is obtained as the collective extension.

In Example 9, however, how an agent performs a subjective evaluation is left as a black-box. The situation is represented using a general AFLP ψ where $\psi = \langle\,(AF_{grd}),\, (LP^1_{stb}, LP^2_{stb}, LP^3_{stb})\,\rangle$ with $AF_{grd} = (\{a,b,c\}, \{(a,b),(b,c)\})$, $LP^1_{stb} = \{\, p \leftarrow not\, q \,\}$, $LP^2_{stb} = \{\, c \leftarrow p, \quad p \leftarrow \}$, and $LP^3_{stb} = \{\, b \leftarrow not\, q \,\}$. Then (AF_{grd}, LP^1_{stb}) has the AFLP extension $\{a,c\}$; (AF_{grd}, LP^2_{stb}) has the AFLP extension $\{a,c\}$; (AF_{grd}, LP^3_{stb}) has the AFLP extension $\{a,b\}$. In this case, the AFLP state of ψ is (Γ) with $\Gamma = (\{\{a,c\}\}, \{\{a,c\}\}, \{\{a,b\}\})$. As such, three agents evaluate the common AF based on their private knowledge base, which results in three individual sets of extensions in the AFLP state. Observe that in this case, the private knowledge of the agents are related to p and q, and only the third agent is *influenced* by his private knowledge base in drawing the conclusion.

When multiple agents argue on the common AF, argument-wise aggregation is characterized using AFLP as follows. Suppose $\Gamma = (T_1, \ldots, T_k)$ $(k \geq 1)$ with $T_i \subseteq 2^A$ where A is the set of arguments of AF. For any $E \subseteq A$, let $\mathscr{F}_\Gamma(E) = h$ where h is the number of occurrences of E in $T_1, \ldots, T_k$. Define $max\mathscr{F}_\Gamma = \{E \mid \mathscr{F}_\Gamma(E) \text{ is maximal}\}$.

Definition 20. Let $\psi = \langle\, \mathbb{AF}^1,\, \mathbb{LP}^m \,\rangle$ $(m \geq 1)$ be a general AFLP that consists of a single AF and multiple LPs. When ψ has the AFLP state (Γ) with $\Gamma = (T_1, \ldots, T_m)$, the *collective extension* by majority voting is any extension in $max\mathscr{F}_\Gamma$.

Applying it to the above example, $max\mathscr{F}_\Gamma = \{\{a,c\}\}$. In Definition 20, if there is $E \subseteq A$ such that $\mathscr{F}_\Gamma(E) = m$, then E is included in every T_i $(1 \leq i \leq m)$. In this case, all agents *agree on* E.

4.3 Multi-context System

Multi-context system (MCS) has been introduced as a general formalism for integrating heterogeneous knowledge bases [4]. An MCS $M = (C_1, \ldots, C_n)$ consists of contexts $C_i = (L_i, kb_i, br_i)$ $(1 \leq i \leq n)$, where $L_i = (KB_i, BS_i, ACC_i)$ is a logic, $kb_i \in KB_i$ is a knowledge base of L_i, BS_i is the set of possible belief sets, $ACC_i : KB_i \mapsto 2^{BS_i}$ is a semantic function of L_i, and br_i is a set of L_i-bridge rules of the form:

$$s \leftarrow (c_1{:}p_1), \ldots, (c_j{:}p_j), \mathit{not}\,(c_{j+1}{:}p_{j+1}), \ldots, \mathit{not}\,(c_m{:}p_m)$$

where, for each $1 \leq k \leq m$, we have that: $1 \leq c_k \leq n$, p_k is an element of some belief set of L_{c_k}, and $kb_i \cup \{s\} \in KB_i$. Intuitively, a bridge rule allows us to add s to a context, depending on the beliefs in the other contexts. Given a rule r of the above form, we denote $head(r) = s$. The semantics of an MCS is described by the notion of belief states. A *belief state* of an MCS $M = (C_1, \ldots, C_n)$ is a tuple $S = (S_1, \ldots, S_n)$ where $S_i \in BS_i$ $(1 \leq i \leq n)$. Given a belief state S and a bridge rule r of the above form, r is *applicable* in S if $p_\ell \in S_{c_\ell}$ for each $1 \leq \ell \leq j$ and $p_k \notin S_{c_k}$ for each $j+1 \leq k \leq m$. By $app(B,S)$ we denote the set of the bridge rules $r \in B$ that are applicable in S. A belief state S of M is an *equilibrium* if $S_i \in ACC_i(kb_i \cup \{head(r) \mid r \in app(br_i, S)\})$ for any i $(1 \leq i \leq n)$.

Given an LPAF $\varphi = \langle LP_\mu, AF_\omega \rangle$, the *corresponding MCS* of φ is defined by $\varphi_{mcs} = (C_1, C_2)$ where $C_1 = (L_1, LP_\mu, br_1)$ in which L_1 is the logic of LP under the μ semantics and $br_1 = \{a \leftarrow (c_2 : a) \mid a \in A\,|_{LP}\}$; and $C_2 = (L_2, AF_\omega, \varnothing)$ where L_2 is the logic of AF under the ω semantics. Intuitively, the bridge rules transfer the acceptability of arguments in AF_ω to LP_μ.

Proposition 8. *Let $\varphi = \langle LP_\mu, AF_\omega \rangle$ be an LPAF framework and φ_{mcs} the* corresponding MCS *of φ. If AF_ω is consistent then (S_1, S_2) is an equilibrium of φ_{mcs} iff S_1 is an LPAF model of φ and S_2 is an ω extension of AF_ω.*

Let $\psi = \langle AF_\omega, LP_\mu \rangle$ be an AFLP framework with $AF_\omega = (A, R)$. The *corresponding MCS* of ψ is defined by $\psi_{mcs} = (C_1, C_2)$ where $C_1 = (L_1, AF_\omega, br_1)$ in which L_1 is the logic of AF under the ω semantics, and $br_1 = \{(y,x) \leftarrow (c_2 : a) \mid \exists a \exists x [a \in A \cap \mathscr{B}_{LP}$ and $(x,a) \in R]\}$ where $y (\notin A)$ is a new argument; $C_2 = (L_2, LP_\mu, \varnothing)$ where L_2 is the logic of LP under the μ semantics. As with LPAF, the bridge rules transfer the acceptability of arguments from LP_μ to AF_ω. We assume that new arguments and attacks introduced by the bridge rules br_1 are respectively added to the set of arguments and attacks of AF.

Proposition 9. *Let $\psi = \langle AF_\omega, LP_\mu \rangle$ be an AFLP framework and ψ_{mcs} the* corresponding MCS *of ψ. If LP_μ is consistent then (S_1, S_2) is an equilibrium of ψ_{mcs} iff $S_1 \setminus Y$ is an AFLP extension of ψ and S_2 is a μ model of LP_μ, where Y is the set of new arguments introduced by br_1.*

A general LPAF $\varphi = \langle \mathbb{LP}_m, \mathbb{AF}_n \rangle$ can be viewed as a collection of MCS. Let C_i^j be the corresponding MCS of $\langle LP^i_{\mu_i}, AF^j_{\omega_j} \rangle$. It is easy to see that by Proposition 8, $(C_i^1, \ldots, C_i^n)$ can be used to characterize the i-th element Σ_i of the LPAF state $(\Sigma_1, \ldots, \Sigma_m)$ of φ. A similar characterization of an AFLP state using MCS could be derived by Proposition 9. A simple LPAF/AFLP is captured as an MCS with a restriction of two systems (Propositions 8 and 9). However, φ_{mcs} (resp. ψ_{mcs}) is well-defined only if its submodule AF_ω (resp. LP_μ) is *consistent*. This is because an MCS assumes that each context is consistent. By contrast, LPAF/AFLP just neglects rules/arguments relying on information that comes from inconsistent AF/LP. As such, LPAF/AFLP shares a view similar to MCS while it is *different* from MCS in general.

4.4 Constrained Argumentation Frameworks

Constrained argumentation frameworks (CAF) [13] could be viewed as another attempt to extend AF with a logical component. A CAF is of the form $\langle A, R, C \rangle$ where (A,R) is an AF and C is a propositional formula over A. A set of arguments S satisfies C if $S \cup \{\neg a \mid a \in A \setminus S\} \models C$. For a semantics ω, an ω *C-extension* of $\langle A, R, C \rangle$ is an ω extension of (A,R) that satisfies C, i.e., the constraint C is used to eliminate undesirable extensions. Therefore, a CAF can be viewed as an LPAF (LP_μ, AF_ω) where AF_ω is the original AF of the CAF and LP_μ is used to verify the condition C.

Consider a CAF $\delta = \langle A, R, C \rangle$. For simplicity of the presentation, assume that C is in DNF. For $a \in A$, let *na* be a unique new atom associated with a, denoting that a is not acceptable. Let $\top$ be a special atom denoting *true*. Define the logic program $LP(C)$ as: $LP(C) = \{\top \leftarrow l'_1, \ldots, l'_n \mid$ a conjunct $l_1 \wedge \cdots \wedge l_n$ is in C and $l'_i = a$ if $l_i = a$, and $l'_i = not\, a$ if $l_i = \neg a\} \cup \{na \leftarrow not\ a, \quad \leftarrow a, na \mid a \in A\} \cup \{\leftarrow not\ \top\}$. We can easily verify that a set of arguments S satisfies C iff $S \cup \{na \mid a \in A \setminus S\} \cup \{\top\}$ is a stable model of $LP(C)$. The next proposition highlights the flexibility of LPAF in that it can also be used to express preferences among extensions of AF.

Proposition 10. *Let $\delta = \langle A, R, C \rangle$ be a CAF. Then, $(LP(C)_{stb}, AF_\omega)$ has an LPAF model M iff $M \setminus (\{na \mid a \in A\} \cup \{\top\})$ is an ω C-extension of δ.*

5 Complexity

The complexity of LPAF/AFLP depends on the complexities of LP and AF. Let us consider the *model existence problem* of simple LPAF frameworks, denoted by $\texttt{Exists}^M$, which is defined as: "*given an LPAF framework φ, determine whether φ has an LPAF model.*" For a simple LPAF framework $\varphi = \langle LP_\mu, AF_\omega \rangle$, the existence of an LPAF model of φ depends on μ and ω. For example, if $\mu =$ *well-founded* and $\omega =$ *grounded* then φ has a unique LPAF model which can be computed in polynomial time (if LP is a normal logic program); on the other hand, if $\mu =$ *stable* and $\omega =$ *stable* then the existence of an LPAF model of φ is not guaranteed. Generally, the next result holds.

Proposition 11. *Let $\varphi = \langle LP_\mu, AF_\omega \rangle$ be a simple LPAF framework such that μ is not universal. Also, let C_μ and C_ω be the complexity classes of LP_μ and AF_ω in the polynomial hierarchy, respectively, and* $\max(C_\mu, C_\omega)$ *the higher complexity class among C_μ and C_ω. Then the model existence problem of φ belongs to the complexity class* $\max(C_\mu, C_\omega)$.

Intuitively, the result follows from the observation that we can guess a pair (X,Y) and check whether Y is an ω extension of AF_ω and X is a μ model of $LP_\mu \cup \{a \leftarrow \mid a \in Y \cap A\mid_{LP}\}$. A similar argument is done for a simple AFLP framework. As an example, the existence of a stable model of a propositional disjunctive LP is in Σ_2^P [9] while the existence of extensions in AF is generally in *NP* or trivial [8], then $\texttt{Exists}^M$ for LPAF/AFLP involving $\mu =$ *stable* is in Σ_2^P where ω is one of the semantics of AF considered in this paper. Other semantics of AF (e.g. semi-stable, ideal, etc.) or LP (e.g. supported, possible models, etc.) can be easily adapted.

The model existence problem of simple LPAF/AFLP can be generalized to the state existence problem of general LPAF/AFLP frameworks, and it can be shown that it is the highest complexity class among all complexity classes involved in the general framework. Similar arguments can be used to determine the complexity class of credulous or skeptical reasoning in LPAF/AFLP. For example, the skeptical entailment in LPAF, i.e., checking whether an atom a belongs to every LPAF model of $\varphi = \langle LP_{stb}, AF_{\omega} \rangle$ is in Π_2^P. We omit detailed discussion for space limitation.

6 Concluding Remarks

Several studies have attempted to integrate LP and AF–translating from one into the other (e.g. [5,6]), or incorporating rule bases into an AF in the context of structured argumentation (e.g. [2,7]). An approach taken in this paper is completely different from those approaches. We do not merge LP and AF while interlinking two components in different manners. LPAF and AFLP enable to combine different reasoning tasks while keeping independence of each knowledge representation. Separation of two frameworks also has an advantage of flexibility in dynamic environments, and several LPs and AFs are freely combined in general LPAF/AFLP frameworks under arbitrary semantics. In addition, it supports an elaboration tolerant use of various knowledge representation frameworks. The potential of the proposed framework is shown by several applications to existing KR frameworks. LPAF or AFLP is realized by linking solvers of LP and AF.

In the proposed framework, LP imports ω extensions from AF in LPAF, while AF imports μ models from LP in AFLP. We can also consider frameworks such that LP (resp. AF) imports *skeptical/credulous consequences* from AF_{ω} (resp. LP_{μ}). Such frameworks are realized by importing the intersection/union of ω extensions of AF to LP (or μ models of LP to AF). In this paper we considered extension based semantics of AF. If we consider the *labelling based semantics* of AF, each argument has three different justification states, *in*, *out*, or *undecided*. In this case, LPAF/AFLP is defined in a similar manner by selecting a 3-valued semantics of logic programs. The current framework can be further extended and applied in several ways. For instance, we can extend it to allow a single LP/AF to refer to multiple AFs/LPs. If AF_{ω} is coupled with a *probabilistic logic program* LP_{μ}, an AFLP (AF_{ω}, LP_{μ}) could be used for computing probabilities of arguments in LP_{μ} and realizing *probabilistic argumentation* in AF_{ω} [12]. As such, the proposed framework has potential for rich applications in AI.

References

1. Baumann, R.: Context-free and context-sensitive kernels: update and deletion equivalence in abstract argumentation. In: Proceedings of the ECAI 2014, pp. 63–68. IOS Press (2014)
2. Besnard, P., Hunter, A.: Constructing argument graphs with deductive arguments: a tutorial. Argum. Comput. **5**(1), 5–30 (2014)
3. Bodanza, G., Tohmé, F., Auday, M.: Collective argumentation: a survey of aggregation issues around argumentation frameworks. Argum. Comput. **8**(1), 1–34 (2017)
4. Brewka, G., Eiter, T.: Equilibria in heterogeneous nonmonotonic multi-context systems. In: AAAI'07, pp. 385–390 (2007)

5. Caminada, M., Sá, S., Alcântara, J., Dvořák, W.: On the equivalence between logic programming semantics and argumentation semantics. J. Approx. Reason. **58**, 87–111 (2015)
6. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and *n*-person games. Artif. Intell. **77**, 321–357 (1995)
7. Dung, P.M., Kowalski, R.A., Toni, F.: Assumption-based argumentation. In: Rahwan, I., Simari, G.R. (Eds.) Argumentation in Artificial Intelligence, pp. 199–218. Springer, Boston (2009). https://doi.org/10.1007/978-0-387-98197-0_10
8. Dvorák, W., Dunne, P.E.: Computational problems in formal argumentation and their complexity. In: Handbook of Formal Argumentation, pp. 631–688. College Publications (2018)
9. Eiter, T., Leone, N., Saccá, D.: Expressive power and complexity of partial models for disjunctive deductive databases. Theor. Comput. Sci. **206**, 181–218 (1998)
10. Eiter, T., Fink, M., Woltran, S.: Semantical characterizations and complexity of equivalences in answer set programming. ACM TOCL **8**(3), 17 (2007)
11. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Proceedings of the JICSLP, pp. 1070–1080. MIT Press (1988)
12. Hunter, A.: A probabilistic approach to modelling uncertain logical arguments. J. Approx. Reason. **54**, 47–81 (2013)
13. Coste-Marquis, S., Devred, C., Marquis, P.: Constrained argumentation frameworks. In: Proceedings of the KR'06, pp. 112–122 (2006)
14. Przymusinski, T.C.: Stable semantics for disjunctive programs. New Gen. Comput. **9**, 401–424 (1991)
15. Sakama, C., Son, T.C.: Interlinking logic programs and argumentation frameworks. In: Proceeding of the NMR-2021, pp. 305–314 (2021). https://sites.google.com/view/nmr2021/home
16. Van Gelder, A., Ross, K., Schlipf, J.S.: The well-founded semantics for general logic programs. J. ACM **38**, 620–650 (1991)