Interlinking Logic Programs and Argumentation Frameworks

Chiaki Sakama[1] and **Tran Cao Son**[2]

[1]Wakayama University, Japan
[2]New Mexico State University, USA

LPNMR 2022@Genova

Background

Logic Program vs. **Argumentation Framework**

	LP	**AF**
knowledge	facts & rules	arguments & attacks
reasoning	commonsense reasoning	argumentative reasoning

- LP and AF specify different types of knowledge and realize different types of reasoning.
- In our daily life, we often use two modes of reasoning **interchangeably**.

Example

Consider a logic program representing knowledge:

$$get\text{-}vaccine \leftarrow safe \land effective$$
$$\neg get\text{-}vaccine \leftarrow not\ safe$$

where we get a vaccine if it is safe and effective.

Example

Consider a logic program representing knowledge:

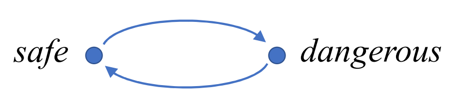
$$get\text{-}vaccine \leftarrow safe \wedge effective$$
$$\neg get\text{-}vaccine \leftarrow not\ safe$$

where we get a vaccine if it is safe and effective.

- To see whether a vaccine is safe and effective, we consult an expert opinion.
- It is often the case that multiple experts have different opinions.

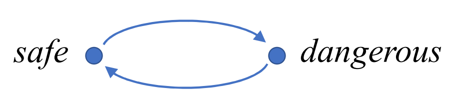
Example

- In this case, we observe argumentation among experts and take it into account to make a decision.
- The truth value of **safe** is determined by an external argumentation framework *AF* such as:



Example

- In this case, we observe argumentation among experts and take it into account to make a decision.
- The truth value of **safe** is determined by an external argumentation framework *AF* such as:



- 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 the chosen semantics** and makes a decision using his/her own LP.

Motivation

The example tells us that

- We need a framework in which a logic program refers to the result of argumentation.
- A logic programming reasoner has the option to choose the semantics of AF as well as the semantics of LP.
- If an argument is not justified in AF, an LP reasoner will not employ the argument.

Motivation

The example tells us that

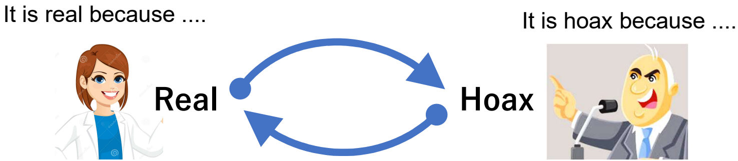
- We need a framework in which a logic program refers to the result of argumentation.
- A logic programming reasoner has the option to choose the semantics of AF as well as the semantics of LP.
- If an argument is not justified in AF, an LP reasoner will not employ the argument.

- AF is transformed to LP, and vice versa, and one could perform both argumentative reasoning and deductive reasoning in a single framework.
- However, the transformational approach requires that two frameworks have the corresponding semantics, i.e., an LP reasoner cannot choose an arbitrary AF semantics.

Example

Consider a debate on whether global warming is occurring.

- Scientists and politicians make different claims based on evidences and scientific knowledge.
- AF is used for representing the debate, while arguments appearing in AF are generated as the results of reasoning from background knowledge of participants.



Motivation

The example tells us that

- We need a framework in which an argumentation framework refers to the result of reasoning in LP.
- An AF participant has the option to choose the semantics of LP as well as the semantics of AF.
- If an argument is not supported in LP, an AF participant will not use the argument.

Motivation

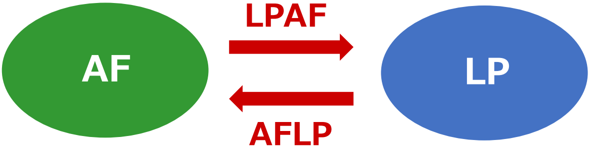
The example tells us that

- We need a framework in which an argumentation framework refers to the result of reasoning in LP.
- An AF participant has the option to choose the semantics of LP as well as the semantics of AF.
- If an argument is not supported in LP, an AF participant will not use the argument.

- Argumentation can have internal structure for reasoning about arguments in structured argumentation.
- However, merging argumentation and knowledge bases into a single framework produces a huge argumentation structure that is complicated and hard to manage.

Purpose

- We introduce new frameworks, called **LPAF** and **AFLP**, for interlinking LPs and AFs.
- **LPAF** uses the result of argumentation in AFs for reasoning in LPs, while **AFLP** uses the result of reasoning in LPs for arguing in AFs.
- **LPAF** and **AFLP** enable to combine different reasoning tasks while keeping independence of each knowledge representation.



LP and AF

- A logic program (LP) is a finite set of rules:

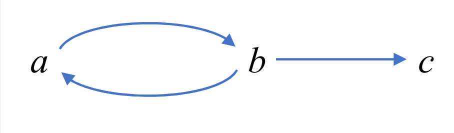
$$p_1 \vee \cdots \vee p_l \leftarrow \; q_1, \ldots, q_m, \, not\, q_{m+1}, \ldots, not\, q_n$$

  where p_i and q_j are propositional variables.
- A logic program LP under the μ semantics is denoted by LP_μ.

- An argumentation framework (AF) is a pair (A, R) where A is a finite set of arguments and $R \subseteq A \times A$ is an attack relation.
- An argumentation framework AF under the ω semantics is denoted by AF_ω.

Example of AF

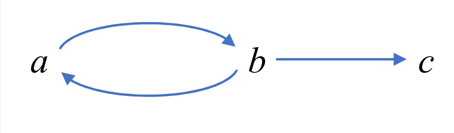
Given $AF = (\{a, b, c\}, \{(a, b), (b, a), (b, c)\})$:



- AF has 2 stable extensions: $\{a, c\}$, $\{b\}$;
- AF has the single grounded extension $\varnothing$.

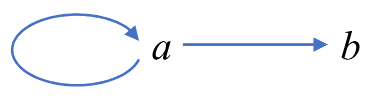
Example of AF

Given $AF = (\{a, b, c\}, \{(a, b), (b, a), (b, c)\})$:



- AF has 2 stable extensions: $\{a, c\}$, $\{b\}$;
- AF has the single grounded extension $\varnothing$.

Given $AF = (\{a, b\}, \{(a, a), (a, b)\})$:



- AF has no stable extension;
- AF has the single grounded extension $\varnothing$.

From AF to LP

Assume that AF and LP share the same propositional language, and no rule in LP has an argument in its head.

Given $AF = (A, R)$, LP is partitioned into $LP = LP^{+A} \cup LP^{-A}$ where $LP^{+A} = \{r \in LP \mid body(r) \cap A \neq \emptyset\}$ and

$$LP^{-A} = \{r \in LP \mid body(r) \cap A = \emptyset\}.$$

- Each rule in LP^{+A} refers to arguments, and each rule in LP^{-A} is free from arguments.
- Argument $a \in A$ is referred to in LP if a appears in LP.
- Define $A|_{LP} = \{ a \in A \mid a \text{ is referred to in } LP \}$.

From AF to LP

Assume that AF and LP share the same propositional language, and no rule in LP has an argument in its head.

Given $AF = (A, R)$, LP is partitioned into $LP = LP^{+A} \cup LP^{-A}$ where $LP^{+A} = \{r \in LP \mid body(r) \cap A \neq \emptyset\}$ and

$$LP^{-A} = \{r \in LP \mid body(r) \cap A = \emptyset\}.$$

- Each rule in LP^{+A} refers to arguments, and each rule in LP^{-A} is free from arguments.
- Argument $a \in A$ is referred to in LP if a appears in LP.
- Define $A|_{LP} = \{a \in A \mid a \text{ is referred to in } LP\}$.

Let $AF = (A, R)$ and $\mathcal{A} \subseteq 2^A$. Then a μ model of LP extended by $\mathcal{A}$ is:

- a μ model of $LP \cup \{a \leftarrow \mid a \in E \cap A|_{LP}\}$ for some $E \in \mathcal{A}$ if $\mathcal{A} \neq \emptyset$;
- a μ model of LP^{-A}, otherwise.

Simple LPAF framework

- A simple LPAF framework is a pair $\varphi = \langle LP_\mu, AF_\omega \rangle$, where LP_μ is an LP under the μ semantics and AF_ω is an AF under the ω semantics.
- When AF_ω has the set of ω extensions $\mathcal{E}^\omega = \{E_1, \ldots, E_k\}$, an LPAF model of φ is defined as a μ model of LP_μ extended by $\mathcal{E}^\omega$, i.e., a μ model of

  $$LP_\mu \cup \{a \leftarrow \mid a \in E_i \cap A|_{LP_\mu}\} \text{ for some } E_i \in \mathcal{E}^\omega$$

  where $A|_{LP_\mu} = \{ a \in A \mid a \text{ appears in } LP_\mu \}$.

Simple LPAF framework

- A simple LPAF framework is a pair $\varphi = \langle LP_\mu, AF_\omega \rangle$, where LP_μ is an LP under the μ semantics and AF_ω is an AF under the ω semantics.
- When AF_ω has the set of ω extensions $\mathcal{E}^\omega = \{E_1, \ldots, E_k\}$, an LPAF model of φ is defined as a μ model of LP_μ extended by $\mathcal{E}^\omega$, i.e., a μ model of

$$LP_\mu \cup \{a \leftarrow \mid a \in E_i \cap A|_{LP_\mu}\} \text{ for some } E_i \in \mathcal{E}^\omega$$

  where $A|_{LP_\mu} = \{ a \in A \mid a \text{ appears in } LP_\mu \}$.
- If $\mathcal{E}^\omega = \emptyset$, an LPAF model is constructed by rules that are free from arguments in AF
  (i.e., no use of rules that contain arguments in AF_ω).

Example

Consider $\varphi_1 = \langle\, LP_{stb},\; AF_{stb}\,\rangle$ where *stb* means *stable*

- $LP_{stb} = \{\, p \leftarrow a, \quad q \leftarrow not\, a\,\}$;
- $AF_{stb} = (\{a, b\}, \{(a, b), (b, a)\})$.
- As AF_{stb} has two stable extensions $\{a\}$ and $\{b\}$, φ_1 has two LPAF models $\{p, a\}$ and $\{q\}$.

Example

Consider $\varphi_1 = \langle\, LP_{stb},\ AF_{stb}\,\rangle$ where *stb* means *stable*

- $LP_{stb} = \{\, p \leftarrow a,\quad q \leftarrow not\, a\,\}$;
- $AF_{stb} = (\{a, b\}, \{(a, b), (b, a)\})$.
- As AF_{stb} has two stable extensions $\{a\}$ and $\{b\}$, φ_1 has two LPAF models $\{p, a\}$ and $\{q\}$.
- If $\omega = grounded$ then AF_{grd} has the single extension $\varnothing$. Then $\langle\, LP_{stb},\ AF_{grd}\,\rangle$ has the single LPAF model $\{q\}$.

Example

Consider $\varphi_1 = \langle\, LP_{stb},\; AF_{stb} \,\rangle$ where *stb* means *stable*

- $LP_{stb} = \{\, p \leftarrow a, \quad q \leftarrow not\, a \,\}$;
- $AF_{stb} = (\{a, b\}, \{(a, b), (b, a)\})$.
- As AF_{stb} has two stable extensions $\{a\}$ and $\{b\}$, φ_1 has two LPAF models $\{p, a\}$ and $\{q\}$.
- If $\omega = grounded$ then AF_{grd} has the single extension $\varnothing$. Then $\langle\, LP_{stb},\; AF_{grd} \,\rangle$ has the single LPAF model $\{q\}$.

Consider $\varphi_2 = \langle\, LP_{stb},\; AF_{stb} \,\rangle$ where

- $LP_{stb} = \{\, p \leftarrow not\, a, \quad q \leftarrow not\, p \,\}$;
- $AF_{stb} = (\{a, b\}, \{(a, b), (a, a)\})$.
- 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 $\{q\}$.

Properties (1)

$\mathbf{M}_\varphi$: the set of LPAF models of φ.

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 $\mathcal{E}^{\omega_1}_{AF^1} \neq \varnothing$. If $\mathcal{E}^{\omega_1}_{AF^1} \subseteq \mathcal{E}^{\omega_2}_{AF^2}$, then $\mathbf{M}_{\varphi_1} \subseteq \mathbf{M}_{\varphi_2}$.

This implies the inclusion relations with the same AF under different semantics: $\mathbf{M}_{\varphi_1} \subseteq \mathbf{M}_{\varphi_2}$ holds for

$$\varphi_1 = \langle\, LP_\mu,\ AF_{prf} \,\rangle \text{ and } \varphi_2 = \langle\, LP_\mu,\ AF_{com} \,\rangle;$$
$$\varphi_1 = \langle\, LP_\mu,\ AF_{stb} \,\rangle \text{ and } \varphi_2 = \langle\, LP_\mu,\ AF_{prf} \,\rangle;$$
$$\varphi_1 = \langle\, LP_\mu,\ AF_{grd} \,\rangle \text{ and } \varphi_2 = \langle\, LP_\mu,\ AF_{com} \,\rangle,$$

where *com*=complete, *prf*=preferred, *stb*=stable, and *grd*=grounded.

Properties (2)

Two programs LP_μ^1 and LP_μ^2 are uniformly equivalent relative to A (denoted $LP_\mu^1 \equiv_u^A LP_\mu^2$) if for any set of non-disjunctive facts $F \subseteq A$, the programs $LP_\mu^1 \cup F$ and $LP_\mu^2 \cup F$ have the same set of μ models (Eiter, et al. 2007).

Let $\varphi_1 = \langle LP_\mu^1, AF_\omega \rangle$ and $\varphi_2 = \langle LP_\mu^2, AF_\omega \rangle$ be two LPAFs such that $\mathcal{E}^\omega \neq \varnothing$. Then, $\mathbf{M}_{\varphi_1} = \mathbf{M}_{\varphi_2}$ if

- $LP_\mu^1 \equiv_u^A LP_\mu^2$, and
- $A|_{LP_\mu^1} = A|_{LP_\mu^2}$

where $AF_\omega = (A, R)$.

General LPAF

A general LPAF framework is defined as a tuple

$$\varphi = \langle \mathcal{LP}^m, \mathcal{AF}^n \rangle$$

where $\mathcal{LP}^m = (LP^1_{\mu_1}, \ldots, LP^m_{\mu_m})$ and $\mathcal{AF}^n = (AF^1_{\omega_1}, \ldots, AF^n_{\omega_n})$. Each $LP^i_{\mu_i}$ is a logic program LP^i under the μ_i semantics and each $AF^j_{\omega_j}$ is an argumentation framework AF^j under the ω_j semantics.

A general LPAF framework is used in a situation such that multiple agents have individual LPs as their private KBs and each agent refers to open AFs.

General LPAF

A general LPAF framework is defined as a tuple

$$\varphi = \langle\, \mathcal{LP}^m, \mathcal{AF}^n \,\rangle$$

where $\mathcal{LP}^m = (LP^1_{\mu_1}, \ldots, LP^m_{\mu_m})$ and $\mathcal{AF}^n = (AF^1_{\omega_1}, \ldots, AF^n_{\omega_n})$. Each $LP^i_{\mu_i}$ is a logic program LP^i under the μ_i semantics and each $AF^j_{\omega_j}$ is an argumentation framework AF^j under the ω_j semantics.

A general LPAF framework is used in a situation such that multiple agents have individual LPs as their private KBs and each agent refers to open AFs.

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 $\langle\, LP^i_{\mu_i}, AF^j_{\omega_j} \,\rangle$.

General LPAF

Given tuples $(S_1, \ldots, S_k)$ and $(T_1, \ldots, T_l)$, define

$$(S_1, \ldots, S_k) \oplus (T_1, \ldots, T_l) = (S_1, \ldots, S_k, T_1, \ldots, T_l).$$

Let $\langle \mathcal{LP}^m, \mathcal{AF}^n \rangle$ be a general LPAF framework.
The LPAF state $(\Sigma_1, \ldots, \Sigma_m)$ of φ is obtained by

$$(\Sigma_1, \ldots, \Sigma_k) \oplus (\Sigma_{k+1}, \ldots, \Sigma_m) \quad (1 \leq k \leq m-1)$$

where $(\Sigma_1, \ldots, \Sigma_k)$ is the LPAF state of $\langle \mathcal{LP}^k, \mathcal{AF}^n \rangle$ and $(\Sigma_{k+1}, \ldots, \Sigma_m)$ is the LPAF state of $\langle \mathcal{LP}^m_{k+1}, \mathcal{AF}^n \rangle$ where $\mathcal{LP}^m_{k+1} = (LP^{k+1}_{\mu_{k+1}}, \ldots, LP^m_{\mu_m})$.

The above 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.

From LP to AF

Assume that no rule in LP has an argument in its body.
$\mathcal{B}_{LP}$: Herbrand base of LP.

Let $AF = (A, R)$ and $M \subseteq \mathcal{B}_{LP}$. Then AF with support M is defined as $AF^M = (A, R')$ where
$R' = R \setminus \{ (x, a) \mid x \in A \text{ and } a \in A \cap M \}$.

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.

From LP to AF

Assume that no rule in LP has an argument in its body.
$\mathcal{B}_{LP}$: Herbrand base of LP.

Let $AF = (A, R)$ and $M \subseteq \mathcal{B}_{LP}$. Then AF with support M is defined as $AF^M = (A, R')$ where
$R' = R \setminus \{ (x, a) \mid x \in A \text{ and } a \in A \cap M \}$.

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.

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

Simple AFLP framework

- A simple AFLP framework is defined as a pair $\psi = \langle AF_\omega, LP_\mu \rangle$.

Simple AFLP framework

- A simple AFLP framework is defined as a pair $\psi = \langle AF_\omega, LP_\mu \rangle$.
- When LP_μ has the set of μ models $\mathcal{M}^\mu$, an AFLP extension of ψ is defined as an ω extension of $AF^M = (A, R')$ where $M \in \mathcal{M}^\mu$ and $R' = R \setminus \{ (x, a) \mid x \in A \text{ and } a \in A \cap M \}$. (i.e., every tuple attacking $a \in M$ is removed from R).

Simple AFLP framework

- A simple AFLP framework is defined as a pair $\psi = \langle AF_\omega, LP_\mu \rangle$.
- When LP_μ has the set of μ models $\mathcal{M}^\mu$, an AFLP extension of ψ is defined as an ω extension of $AF^M = (A, R')$ where $M \in \mathcal{M}^\mu$ and $R' = R \setminus \{ (x, a) \mid x \in A \text{ and } a \in A \cap M \}$. (i.e., every tuple attacking $a \in M$ is removed from R).
- If $\mathcal{M}^\mu = \emptyset$, an AFLP extension of ψ is an ω extension of (A', R') where $A' = A \setminus \mathcal{B}_{LP}$ and $R' = R \cap (A' \times A')$. (i.e., no use of arguments that rely on LP).

Example

Consider $\psi_1 = \langle\, AF_{stb},\ LP_{stb}\,\rangle$ where

- $AF_{stb} = (\{a, b\}, \{(a, b), (b, a)\})$;
- $LP_{stb} = \{\, a \leftarrow p, \quad p \leftarrow not\, q, \quad q \leftarrow not\, p\,\}$.
- LP_{stb} has two stable models $M_1 = \{a, p\}$ and $M_2 = \{q\}$, then $AF_\omega^{M_1} = (\{a, b\}, \{(a, b)\})$ and $AF_\omega^{M_2} = AF_\omega$.
  As a result, ψ_1 has two AFLP extensions $\{a\}$ and $\{b\}$.

Example

Consider $\psi_1 = \langle\, AF_{stb},\ LP_{stb}\,\rangle$ where

- $AF_{stb} = (\{a, b\}, \{(a, b), (b, a)\})$;
- $LP_{stb} = \{\, a \leftarrow p, \quad p \leftarrow not\, q, \quad q \leftarrow not\, p\,\}$.
- LP_{stb} has two stable models $M_1 = \{a, p\}$ and $M_2 = \{q\}$, then $AF_\omega^{M_1} = (\{a, b\}, \{(a, b)\})$ and $AF_\omega^{M_2} = AF_\omega$.
  As a result, ψ_1 has two AFLP extensions $\{a\}$ and $\{b\}$.
- If we use $\omega = grounded$, then $\langle\, AF_{grd},\ LP_{stb}\,\rangle$ has two AFLP extensions $\{a\}$ and $\emptyset$.

Example

Consider $\psi_1 = \langle\, AF_{stb},\ LP_{stb} \,\rangle$ where

- $AF_{stb} = (\{a, b\}, \{(a, b), (b, a)\})$;
- $LP_{stb} = \{\, a \leftarrow p, \quad p \leftarrow not\, q, \quad q \leftarrow not\, p \,\}$.
- LP_{stb} has two stable models $M_1 = \{a, p\}$ and $M_2 = \{q\}$, then $AF_\omega^{M_1} = (\{a, b\}, \{(a, b)\})$ and $AF_\omega^{M_2} = AF_\omega$. As a result, ψ_1 has two AFLP extensions $\{a\}$ and $\{b\}$.
- If we use $\omega = grounded$, then $\langle\, AF_{grd},\ LP_{stb} \,\rangle$ has two AFLP extensions $\{a\}$ and $\varnothing$.

Consider $\psi_2 = \langle\, AF_{grd},\ LP_{stb} \,\rangle$ where

- $AF_{grd} = (\{a, b, c\}, \{(a, b), (b, c)\})$;
- $LP_{stb} = \{\, a \leftarrow p, \quad p \leftarrow not\, p \,\}$.
- As LP_{stb} has no stable model, ψ_2 has the AFLP extension $\{b\}$ as the grounded extension of $(\{b, c\}, \{(b, c)\})$.

Properties (1)

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 $\mathcal{M}^{\mu_1}_{LP^1} \neq \emptyset$. If $\mathcal{M}^{\mu_1}_{LP^1} \subseteq \mathcal{M}^{\mu_2}_{LP^2}$, then $\mathbf{E}_{\psi_1} \subseteq \mathbf{E}_{\psi_2}$.

Properties (2)

(Baumann 2014) Given $AF^1_\omega = (A_1, R_1)$ and $AF^2_\omega = (A_2, R_2)$,

- AF^1_ω and AF^2_ω are normal deletion equivalent (denoted $AF^1_\omega \equiv_{nd} AF^2_\omega$) 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 $A'_2 = A_2 \setminus A$.
- AF^1_ω and AF^2_ω are local deletion equivalent (denoted $AF^1_\omega \equiv_{ld} AF^2_\omega$) 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.

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

- $\mathcal{M}^\mu = \varnothing$ and $AF^1_\omega \equiv_{nd} AF^2_\omega$; or
- $\mathcal{M}^\mu \neq \varnothing$ and $AF^1_\omega \equiv_{ld} AF^2_\omega$.

General AFLP

A general AFLP framework is defined as a tuple

$$\psi = \langle \mathcal{AF}^n, \mathcal{LP}^m \rangle$$

where $\mathcal{AF}^n = (AF^1_{\omega_1}, \ldots, AF^n_{\omega_n})$ and $\mathcal{LP}^m = (LP^1_{\mu_1}, \ldots, LP^m_{\mu_m})$. 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.

General AFLP

A general AFLP framework is defined as a tuple

$$\psi = \langle\, \mathcal{AF}^n, \mathcal{LP}^m \,\rangle$$

where $\mathcal{AF}^n = (AF^1_{\omega_1}, \ldots, AF^n_{\omega_n})$ and $\mathcal{LP}^m = (LP^1_{\mu_1}, \ldots, LP^m_{\mu_m})$. 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.

An **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 $\langle\, AF^j_{\omega_j},\ LP^i_{\mu_i} \,\rangle$.

General AFLP

A general AFLP has the modularity property.

Let $\psi = \langle\, \mathcal{AF}^n, \mathcal{LP}^m \,\rangle$ be a general AFLP framework. Then the AFLP state $(\Gamma_1, \ldots, \Gamma_n)$ of ψ is obtained by

$$(\Gamma_1, \ldots, \Gamma_k) \oplus (\Gamma_{k+1}, \ldots, \Gamma_n) \quad (1 \leq k \leq n-1)$$

where $(\Gamma_1, \ldots, \Gamma_k)$ is the AFLP state of $\psi_1 = \langle\, \mathcal{AF}^k, \mathcal{LP}^m \,\rangle$ and $(\Gamma_{k+1}, \ldots, \Gamma_n)$ is the AFLP state of $\psi_2 = \langle\, \mathcal{AF}^n_{k+1}, \mathcal{LP}^m \,\rangle$ where $\mathcal{AF}^n_{k+1} = (AF^{k+1}_{\omega_{k+1}}, \ldots, AF^n_{\omega_n})$.

Bidirectional LPAF

A simple bidirectional LPAF framework is defined as a pair $\langle\!\langle\, LP_\mu,\ AF_\omega \,\rangle\!\rangle$.

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 $\mathbf{E}_\psi$.

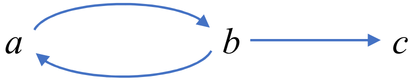
The definition is extended to general bidirectional LPAF.

Example

Consider $\zeta = \langle\!\langle\, LP_{stb},\ AF_{stb}\,\rangle\!\rangle$ where

- $LP_{stb} = \{\, a \leftarrow not\, p, \quad q \leftarrow c\,\}$;
- $AF_{stb} = (\{a, b, c\}, \{(a, b), (b, a), (b, c)\})$.

First, the simple AFLP framework $\langle\, AF_{stb}, LP_{stb}\,\rangle$ has the single AFLP extension $E = \{a, c\}$. Then, the BDLPAF model of ζ becomes $\{a, c, q\}$.



Bidirectional AFLP

A simple bidirectional AFLP framework is defined as a pair $\langle\!\langle\, AF_\omega,\ LP_\mu \,\rangle\!\rangle$.

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 $\mathbf{M}_\varphi$.

The definition is extended to general bidirectional AFLP.

Example

Consider $\eta = \langle\!\langle\, AF_{grd},\ LP_{stb} \,\rangle\!\rangle$ where

- $AF_{grd} = (\{a, b\}, \{(a, b), (b, a)\})$;
- $LP_{stb} = \{\, p \leftarrow a, \quad q \leftarrow not\, a, \quad b \leftarrow q \,\}$.

First, the simple LPAF framework $\langle\, LP_{stb}, AF_{grd} \,\rangle$ has the single LPAF model $M = \{b, q\}$. Then, the BDAFLP extension of η becomes $\{b\}$.

Property

- Given AF_ω and LP_μ, a series of BDLPAF models or BDAFLP extensions can be build by repeatedly referring with each other.
- Starting with the AFLP extensions $\mathbf{E}_\psi^0$, the BDLPAF models $\mathbf{M}_\varphi^1$ extended by $\mathbf{E}_\psi^0$ are produced.
- Then the BDAFLP extensions $\mathbf{E}_\psi^1$ supported by $\mathbf{M}_\varphi^1$ are produced, which in turn produce the BDLPAF models $\mathbf{M}_\varphi^2$ extended by $\mathbf{E}_\psi^1$, and so on.

Property

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

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

Final Remarks

- The **complexity** of LPAF/AFLP depends on the complexities of LP and AF. For instance, the model existence problem of a simple LPAF belongs to the complexity class $\max(C_\mu, C_\omega)$, where C_μ and C_ω are the complexity classes of LP_μ and AF_ω, respectively.
- LPAF/AFLP are applied to several KR frameworks such as **deductive argumentation, argument aggregation, multi-context system**, and **constrained argumentation frameworks**.
- If AF is coupled with **probabilistic LP**, an AFLP could be used for computing probabilities of arguments in LP and realizing **probabilistic AF**.