

プリファレンス付き議論フレームワーク PAF の ASP による議論意味論計算

永吉 信雄¹

若木 利子²

芝浦工業大学 システム工学部 電子情報システム学科*

1 はじめに

人間が行う議論の理論化・形式化を行う研究分野では、1995年 Dung により「議論フレームワーク」(Argumentation Framework, 以後 AF) とその意味論 (Semantics) が提案され [2], 以来電子商取引等の交渉を行うマルチエージェントシステムに 응용が試みられている。また例えば法廷では「個人の幸福追求」より「公共の福祉」を優先する, あるいは商取引では「価格の安さ」より「品質」を優先するなど, 優先情報はしばしば用いられている。そこで2002年 Amgoud により, 優先情報を扱えるよう Dung の議論意味論を拡張した「プリファレンス付き議論フレームワーク」(Preference-based AF, 以後 PAF) とその意味論が提案された。しかし昨年の IJCAI2009 の国際会議で, Amgoud 自身により「2002年に提案した意味論では正しい議論結果が得られない」(勝利した意見の集合である Extension に矛盾が含まれる, 即ち conflict-free 性の非保証) という技術的問題点が指摘された [1]。そこで昨年, ArgMAS2010 で若木により conflict-free を保証する PAF の新たな意味論が提案された [4]。本研究では, 若木の PAF の意味論に基づいた議論計算エンジンを, 解集合プログラミング (以後 ASP) [3] の技術を適用し, 開発した。

2 PAF

2.1 議論フレームワーク AF

Dung の AF は, 議論の場に提出される論証 (意見) とそれらの間の攻撃関係から, 議論で勝利する論証の集合である Extension を求めるモデルである。AF は, $AF = (AR, def)$ で定義される。意味論は4種 (Complete, Stable, Preferred, Grounded) ある (詳細は省略, 文献 [2] 参照)。

2.2 PAF とその意味論の定義

定義 1 (PAF) [4] PAF を

$$PAF = (AR, def, \leq)$$

で定義する。AR は論証集合, def は AR 上の二項関係である。A, B ∈ AR について (A, B) ∈ def, 即ち A def B は「A が B を攻撃する」ことを意味する。≤ は AR 上の二項関係で反射律・推移律を満たす preorder である。(A, B) ∈ ≤, 即ち A ≤ B は「B は A より優先される」ことを意味する。A ≤ B かつ B ≰ A のとき A < B と書く。PAF の意味論は定義 3 で与えられる。定義 2 (Extension 間の優先関係) [4] Dung の意味論における任意の意味論名を Sname で表す。Sname Extension の集合を \mathcal{E}_{AF} とし, \mathcal{E}_{AF} 上の優先関係 \sqsubseteq_{ex} を以下のように定義する。

1. $\forall E, E' \in \mathcal{E}_{AF}$ について次の条件を両方満たす $y \in E'/E$ が存在するなら $E \sqsubseteq_{ex} E'$ である。(図 1)

- $\exists x \in E/E'$ such that $x \leq y$
- $\nexists z \in E/E'$ such that $y < z$

2. \sqsubseteq_{ex} は反射律と推移律を満たす。

$E, E' \in \mathcal{E}_{AF}$ について $E \sqsubseteq_{ex} E'$ は「E' は E より優先される」ことを意味する。

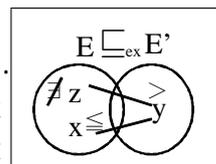


図 1: \sqsubseteq_{ex} の条件

$E \sqsubseteq_{ex} E'$ かつ $E' \not\sqsubseteq_{ex} E$ のとき $E \sqsubset_{ex} E'$ と書く。定義 3 (P-Extension) [4] 所与の PAF において, $AF = (AR, def)$ の Sname Extension の集合を \mathcal{E}_{AF}

とする。E ∈ \mathcal{E}_{AF} が Sname P-Extension であることの必要十分条件は, $\forall E' \in \mathcal{E}_{AF}$ について $E \sqsubseteq_{ex} E'$ なる E' が存在しないことである。

2.3 例題

移動手段について, 車, 電車, 自転車の3つから選びたい。議論の場に出された論証を次に示す。

- C_0 : 電車・自転車を使わないなら車を使う
- C_1 : 車を使えば荷物を多く積める
- C_2 : 車を使えば疲れない
- T_0 : 車・自転車を使わないなら電車を使う
- T_1 : 電車を使えば渋滞に巻き込まれない
- T_2 : 電車を使えば疲れない
- B_0 : 車・電車を使わないなら自転車を使う
- B_1 : 自転車を使えば運動になる
- B_2 : 自転車を使えば渋滞に巻き込まれない

これらの論証間の攻撃関係を図示した AF のグラフを図 2 に示す。矢印は, その根元にある論証が矢印の先の論証を攻撃していることを意味する (図 2 では双方向矢印)。

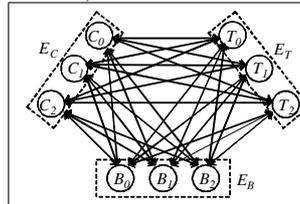


図 2: 論証間の攻撃関係

この攻撃関係は「車を使った場合は自転車を使えない」等, 各選択肢が排他的なことから導かれる。Dung の意味論に基づいてこの AF の議論計算を行うと, Preferred (Stable) Extension として

$$E_C = \{C_0, C_1, C_2\}, E_T = \{T_0, T_1, T_2\}, E_B = \{B_0, B_1, B_2\}$$

これらが求められる。これらはそれぞれの見方でどれも勝利した意見の集合であり, 優劣はつけがたい。ここで「運動するよりも疲れたくない」「荷物を多く積めるよりも渋滞に巻き込まれたくない」という要望 (プリファレンス) をエージェントが持っている時,

$$B_1 \leq T_2, B_1 \leq C_2, C_1 \leq T_1, C_1 \leq B_2$$

として表現できる。すると Extension 同士の優先関係は

$$E_B \sqsubseteq_{ex} E_T, E_C \sqsubseteq_{ex} E_T, E_C \sqsubseteq_{ex} E_B, E_B \sqsubseteq_{ex} E_C$$

(図 3), よって Preferred (Stable) P-Extension は唯一 E_T のみとなり, 人間の意図する自然な結果が得られる。

3 ASP による PAF の議論意味論計算

3.1 変換論理プログラムの構成

ASP プログラムのクラスは NDP (Normal Disjunctive Program) 即ち以下の形式のルールの集合である。

$A_1 | \dots | A_k \leftarrow B_1, \dots, B_m, \text{not } B_{m+1}, \dots, \text{not } B_{m+n}$ 但し $0 \leq k, m, n$, また A, B はアトムである。特に $k \leq 1$ のとき NLP (Normal Logic Program) と呼ぶ。NLP, NDP の宣言的意味は解集合 (安定モデル) [3] で与えられる。

PAF の議論計算では, このような ASP プログラムを以下のように複数組み合わせる。

(PAF の議論計算手順)

1. NLP $TrAF$ を用いて AF の Sname Extension の集合 \mathcal{E}_{AF} を計算する
2. NLP $TrPref$ を用いて \mathcal{E}_{AF} 上の優先関係 \sqsubseteq_{ex} を生成する
3. NLP $FindMax$ を用いて優先関係 \sqsubseteq_{ex} に関して極大な P-Extension を求める
4. NDP のルール集合により, 所与の論証の sceptical (credulous) な正当化判定を計算 (詳細は省略)。

Computing Semantics of Preference-based Argumentation Frameworks in Answer Set Programming

¹Nobuo Nagayoshi, ²Toshiko Wakaki

*Shibaura Institute of Technology

(1) (NLP $TrAF$ による \mathcal{E}_{AF} の計算)

文献 [6, 7] に従って, 所与の $AF = (AR, def)$ と指定の Sname 意味論から構成される NLP を $TrAF[AR, def; Sname]$ とする. この NLP の解集合 S について以下の定理が成り立つ.

定理 1 [6, 7] E が AF の Sname Extension であるならば, $TrAF[AR, def; Sname]$ は $S \cap \mathcal{I} = \{in(a) \mid a \in E\}$ なる解集合 S を持ち, その逆も成り立つ. 但し $\mathcal{I} = \{in(a) \mid a \in AR\}$.

(2) (NLP $TrPref$ による \sqsubseteq_{ex} の計算)

$PAF = (AR, def, \leq)$ と Sname 意味論名, 及び $AF = (AR, def)$ の Sname Extension $E \in \mathcal{E}_{AF}$ が与えられた時, 定義 4 に従って NLP $TrPref$ を構成する. この NLP の解集合 M から, 定理 2 より, E より優先される Extension $E' \in \mathcal{E}_{AF}$ (即ち $E \sqsubseteq_{ex} E'$) が得られるので, 全ての $E \in \mathcal{E}_{AF}$ について $TrPref$ を構成, 解集合を計算することで, AF の Extension 間の Preference 関係 \sqsubseteq_{ex} を生成できる. なお NLP $TrPref$ と NLP $FindMax$ の構成では, 文献 [5] の解集合間の優先関係 \sqsubseteq を算出する手法を, PAF 議論計算の Extension 間の優先関係 \sqsubseteq_{ex} の算出に応用している.

定義 4 NLP $TrPref$ を以下で定義する.

$$TrPref[AR, def, \leq; Sname, E] \stackrel{def}{=} TrAF[AR, def; Sname] \cup \Gamma_{PREF} \cup \Pi_{PREF}$$

但し Γ_{PREF} は AR, \leq, E から構成される domain dependent な 1.2 のルール, Π_{PREF} は domain independent な 3~12 のルールから成る.

1. 任意の $(a, b) \in \leq$ について, $\leq(a, b) \leftarrow$.
2. $a \in E$ について, $m_1(a) \leftarrow$.
3. $m_2(x) \leftarrow in(x)$.
4. $\leq(x, x) \leftarrow$.
5. $\leq(x, z) \leftarrow \leq(x, y), \leq(y, z)$.
6. $\leq(y, z) \leftarrow \leq(x, y), not \leq(y, x)$.
7. $gr_1(x, y) \leftarrow m_1(x), \leq(x, y), m_2(y), not m_2(x), not m_1(y)$.
8. $gr_2(y, z) \leftarrow m_2(y), \leq(y, z), m_1(z), not m_1(y), not m_2(z)$.
9. $attacked(y) \leftarrow gr_2(y, z)$.
10. $defeated(x) \leftarrow gr_1(x, y), not attacked(y)$.
11. $better \leftarrow defeated(x)$.
12. $\leftarrow not better$.

定理 2 $E \in \mathcal{E}_{AF}$ に対して, $E \sqsubseteq_{ex} E'$ なる AF の Sname Extension $E' \in \mathcal{E}_{AF}$ が存在するならば, $M \cap \mathcal{I} = \{in(a) \mid a \in E'\}$ なる $TrPref[AR, def, \leq; Sname, E]$ の解集合 M が存在する. 逆に, $TrPref[AR, def, \leq; Sname, E]$ の解集合 M が存在するならば, M から構成される $E' \stackrel{def}{=} \{a \mid in(a) \in M\}$ について, $E \sqsubseteq_{ex} E'$ かつ $E' \in \mathcal{E}_{AF}$ が成り立つ.

(3) (NLP $FindMax$ による P-Extension の計算)

$AF = (AR, def)$ の Sname Extension の集合 \mathcal{E}_{AF} と \mathcal{E}_{AF} 上の Preference 関係 \sqsubseteq_{ex} が与えられた時, 定義 5 に従って NLP $FindMax$ を構成する. この NLP の解集合 U から定理 3 に基づき, AF の Extension のうち Preference 関係で極大なもの, 即ち P-Extension が求められる.

定義 5 NLP $FindMax$ を以下で定義する.

$$FindMax[\mathcal{E}_{AF}, \sqsubseteq_{ex}] \stackrel{def}{=} \Psi \cup \Sigma \cup \{ex(i) \leftarrow \cdot \mid E_i \in \mathcal{E}_{AF}, 1 \leq i \leq |\mathcal{E}_{AF}|\}$$

但し $\Sigma = \{\sqsubseteq(i, j) \leftarrow \cdot \mid E_i \sqsubseteq_{ex} E_j\}$, Ψ は domain independent な 1~5 のルールから成る.

1. $\sqsubseteq(x, x) \leftarrow ex(x)$.
2. $\sqsubseteq(x, z) \leftarrow \sqsubseteq(x, y), \sqsubseteq(y, z)$.
3. $\sqsubseteq(x, y) \leftarrow \sqsubseteq(x, y), not \sqsubseteq(y, x)$.
4. $worse(x) \leftarrow \sqsubseteq(x, y)$.
5. $p - ex(x) \leftarrow ex(x), not worse$.

定理 3 $FindMax[\mathcal{E}_{AF}, \sqsubseteq_{ex}]$ の解集合を U とする. この時, $p - ex(i) \in U$ ならば $E_i \in \mathcal{E}_{AF}$ は PAF の Sname P-Extension であり, その逆も成り立つ.

3.2 議論計算手続き

ASP を用いた PAF の議論意味論計算手続き CompPAF を以下に示す.

手続き: $CompPAF(AR, def \leq, Sname, [Ag, se/ce], \mathcal{E}_{PAF})$

入力: $PAF = (AR, def, \leq)$, Sname 意味論の指定.

質問応答を行う場合は質問の論証 Ag と評価方法 (credulous または sceptical) の指定.

出力: $PAF = (AR, def, \leq)$ のすべての Sname P-Extension の集合 \mathcal{E}_{PAF} を出力.

質問応答を行う場合は正当化判定の結果を出力.

```

step0.0.  $\leq = \emptyset$  なら [7] の手続きを用いて計算, return.
step1.0.  $TrAF[AR, def; Sname]$  の解集合の集合  $\mathcal{AS}$  を計算する.
    1.1. 各  $S \in \mathcal{AS}$  に自然数  $i$  を割り当てる. これを  $S_i$  と表す. 但し  $1 \leq i \leq |\mathcal{AS}|$ 
    1.2. すべての  $S_i$  について  $E_i = \{a \mid in(a) \in S_i\}$  を求め,  $AF = (AR, def)$  の Sname Extension の集合  $\mathcal{E}_{AF}$  に格納.  $i$  を Extension ID と呼ぶとする.
step 2.0.  $\Sigma := \emptyset$ 
    2.1. 任意の  $E_i \in \mathcal{E}_{AF}$  について  $TrPref[AR, def, \leq; Sname, E_i]$  の解集合を計算. 無矛盾ならその全ての解集合  $M$  について 2.1., 2.2. を行う.
    2.1.  $E' = \{a \mid in(a) \in M\}$  の Extension ID:  $j$  を求める.
    2.2.  $\Sigma := \Sigma \cup \{\sqsubseteq(i, j) \leftarrow \cdot \mid E_i \sqsubseteq_{ex} E' \text{ 但し } E' \text{ の ID は } j\}$ 
step3.0.  $FindMax[\mathcal{E}_{AF}, \sqsubseteq_{ex}]$  の解集合  $U$  を計算する.  $\Omega := \{i \mid p - ex(i) \in U\}$  として, Sname P-Extension の集合  $\mathcal{E}_{PAF}$  を次のように計算して, 出力する. 但し  $\Omega$  は Sname P-Extension の Extension ID の集合である.
     $\mathcal{E}_{PAF} \stackrel{def}{=} \{E_i \mid i \in \Omega \text{ について Extension ID: } i \text{ の Sname Extension } E_i \in \mathcal{E}_{AF}\}$ 
step4.0. 論証  $Ag$  の正当化判定の計算 (詳細は紙面の都合上省略)
    
```

3.3 実装・検証

PAF の議論計算プログラム CompPAF を ASP ソルバの DLV と C 言語を用いて実装した. PAF を記述したファイルを入力, 意味論等を引数で指定し, PAF の議論計算を行い, 結果を出力する. 例を用いて検証を行い, 正しく計算されていることを確認した. 2.3 で取り上げた例題の実行例を図 4 に示す. 図 4 より, \leq の情報を記述したファイル $ex_pref.txt$ を読み込まない場合は, $AF = (AR, def)$ の Preferred Extension が 3 つ, 読み込めると $PAF = (AR, def, \leq)$ の Preferred P-Extension E_T が出力されていることがわかる.

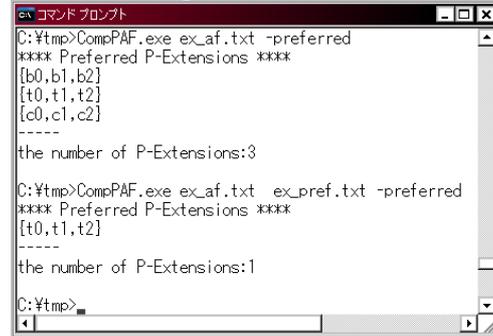


図 4: 実行例

4 おわりに

本研究では PAF の議論意味論計算を ASP の技術を用いて実現する方法を提案し, 提案手法を実装した. 本研究で作成した議論計算プログラム CompPAF は, 近々 Web 上に公開予定である.

参考文献

- [1] L. Amgoud, S. Vesic: Repairing Preference-Based Argumentation Frameworks, International Joint Conference on Artificial Intelligence, Proc. of IJCAI 2009, pp.665-670, 2009.
- [2] P. M. Dung: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artificial Intelligence, 77, pp.321-357, 1995.
- [3] M. Gelfond, V. Lifschitz: Classical Negation in Logic Programs and Disjunctive Databases, New Generation Computing 9, pp. 365-385, 1991
- [4] T. Wakaki: Preference-based Argumentation Capturing Prioritized Logic Programming, Proc. of ArgMAS 2010, pp.211-228, 2010
- [5] T. Wakaki, K. Inoue, C. Sakama, K. Nitta: Computing preferred answer sets in answer set programming, Proc of LPAR 2003, LNAI, Vol.2850, Springer-Verlag, pp. 259-273, 2003.
- [6] T. Wakaki, K. Nitta: Computing Argumentation Semantics in Answer Set programming, New Frontiers in Artificial Intelligence, LNAI Vol. 5447, pp. 254-269, Springer, Verlag, 2008.
- [7] 伊藤 浩太, 若水 希子: 解集合プロダクティングによる議論の意味論の計算, 情報処理学会第 70 回全国大会講演論文集, 3V-9, pp.2-253-2-254, 2008.