

解集合プログラミングによるアブダクションを用いた発想的議論の計算

淡嶋 綾子¹原田 雅史²若木 利子³

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

分散コンピュータネットワーク上のマルチエージェントに人間のような議論を行わせるために 1995 年に Dung によって議論の意味論と議論の判定方法が提案された [1]. ここでは真である事実的知識だけを想定しているが, 人間はその他に仮説的知識を考慮した議論も行う. 例えば法廷で検事が "A が犯人である" と主張したいが証拠が足りずに主張を正当化できない場合がある. このとき検事は真偽が不明な仮説も持っており, 自身の所望の主張を勝たせる為に必要な仮説はどれかを知りたい. このような議論における所望の仮説の集合 (説明) を求める為に, 発想的議論の定義と, アブダクションに基づく解集合プログラミングにより発想的議論を計算する方法が提案された [5]. またマルチエージェント間の交渉などにおいては極小の説明が欲しいというニーズがある [6]. そこで本研究では極小説明を算出する方法を提案し, それを文献 [5] の方法に含めた提案手法を実現するツールを実装した. 更に種々の例題を使ってツールの検証を行い, 正しい結果が得られることを確認した.

2 演繹的議論

以下に演繹的議論について述べる [2].

定義 1 (拡張論理プログラム) 拡張論理プログラム (以下 ELP と略記) は以下の形のルール r の集合である.

$$L \leftarrow L_1, \dots, L_m, \text{not } L_{m+1}, \dots, \text{not } L_{m+n}$$

ここで L, L_i をリテラル (アトム A , または $\neg A$) とし, $\leftarrow$ の左側を頭部 ($\text{head}(r)$), 右側を本体と呼ぶ.

定義 2 (論証 [2]) 所与の ELP P において, 論証は P のルールの有限系列 $[r_1, \dots, r_k]$ であり, $\text{head}(r_1)$ を論証 Ag の claim といい $\text{claim}(Ag)$ で表す. また P から作られる極小論証の集合を Args_P で表す.

定義 3 (攻撃関係 [2]) 所与の ELP P における攻撃関係 attacks_P は攻撃の概念 (undercut , rebut 等) によって定義された Args_P 上の二項関係である.

定義 4 (議論フレームワーク) ELP P に関する論証集合 Args_P と Args_P 上の攻撃関係 attacks_P の対 $(\text{Args}_P, \text{attacks}_P)$ を議論フレームワークと呼び, AF_P で表す.

定義 5 (議論の意味論 [1]) 所与の議論フレームワーク $\text{AF} = (Ar, \text{def})$ において, $S \subseteq Ar$ を conflict free な論証集合, $F : 2^{Ar} \rightarrow 2^{Ar}$ を $F(S) = \{a \mid a \text{ は } S \text{ に関して acceptable}\}$ なる関数とすると, 各意味論は complete , grounded , preferred , stable の extension (略して $c.e.$, $g.e.$, $p.e.$, $s.e.$) によって与えられる.

(1) $\text{complete semantics}$: E は $c.e.$ iff $E = F(E)$.

(2) $\text{preferred/grounded semantics}$: E は $g.e.$ ($p.e.$) iff E が $\subseteq$ に関して極大 (極小) の $c.e.$ である.

(3) stable semantics : E は $s.e.$ iff E が $Ar \setminus E$ 内の全ての論証を攻撃している.

定義 6 (Skeptical/Credulous に正当化) (Ar, def) を AF , E_i を指定の意味論の extension とする. この時論証 $a \in E_i$ について,

- 任意の (ある) E_j について $a \in E_j$ ならば, a は skeptical (credulous) に正当化される.

<演繹的議論の問題> ELP P , 議題 G , 意味論 $Sname$ が与えられた時, $Sname$ の意味論において, $\text{claim}(Ag) = G$ なる論証 $Ag \in \text{Args}_P$ が $\text{AF}_P = (\text{Args}_P, \text{attacks}_P)$ に関して正当化できるかを判定せよ.

Computing Minimal Explanation of Abductive Argumentation in Answer Set Programming

¹Ayako Awashima, ²Masashi Harada, ³Toshiko Wakaki

*Shibaura Institute of Technology

3 発想的議論

3.1 発想的議論

以下に発想的議論について述べる. 本研究ではエージェントは発想的プログラム $\langle P, H \rangle$ を持つと仮定する.

定義 7 (発想的プログラム) 発想的プログラムとは $\langle P, H \rangle$ の対である. P は ELP, H は基礎リテラルの集合 (仮説集合) であり, P の各ルールの頭部は H には含まれない.

定義 8 (発想的議論フレームワーク) 発想的プログラム $\langle P, H \rangle$ の下で $\text{AF}_{PUE} = (\text{Args}_{PUE}, \text{attacks}_{PUE})$ (但し $E \subseteq H$) が冗長でない時, これを発想的議論フレームワークと呼び AAF_{PUE} で表す. (冗長については参考文献 [5] を参照)

定義 9 (説明/反説明) 発想的プログラム $\langle P, H \rangle$, $\text{AAF}_{PUE} = (\text{Args}_{PUE}, \text{attacks}_{PUE})$ (但し $E \subseteq H$), 議題 G , 指定の意味論 $Sname$ が与えられた時, $Sname$ 意味論の下で $\text{claim}(Ag) = G$ なる論証 $Ag \in \text{Args}_{PUE}$ が AAF_{PUE} に関して skeptical に正当化される (または credulous に正当化されない) ならば, E は Ag の skeptical な説明 (または credulous な反説明) である.

<発想的議論の問題>

発想的プログラム $\langle P, H \rangle$, 議題 G , 意味論 $Sname$ が与えられた時, $Sname$ の意味論において $\text{claim}(Ag) = G$ なる論証 $Ag \in \text{Args}_{PUE}$ が $\text{AAF}_{PUE} = (\text{Args}_{PUE}, \text{attacks}_{PUE})$ (但し $E \subseteq H$) に関して skeptical に正当化される (されない) ような説明 (反説明) E を求めよ.

3.2 発想的議論の計算

定義 10 (アブダクションと説明 [4]) 発想的プログラム $\langle T, H \rangle$, 基礎リテラルからなる観測 G が与えられた時, E が $\langle T, H \rangle$ に関して G の skeptical (または credulous) な説明であるのは

1. G は $T \cup E$ のすべての (またはいくつかの) 解集合に含まれる.
2. $T \cup E$ が無矛盾であり, $E \subseteq H$

観測 G に対する反説明については文献 [4] を参照. 所与の $\langle P, H \rangle$ から T を以下に示す変換論理プログラムで構成することにより, 定義 10 を用いて発想的議論における説明を求める方法が文献 [5] で提案された.

定義 11 (変換論理プログラム [5]) $\langle P, H \rangle$ と攻撃の概念が与えられた時, 各意味論に対する変換論理プログラムは以下の ELP によって定義される.

- $\text{tr}[\text{Args}_{P \cup H}, \text{attacks}_{P \cup H}; \text{complete}] = \Pi \cup \Gamma \cup \Xi$
- $\text{tr}[\text{Args}_{P \cup H}, \text{attacks}_{P \cup H}; \text{preferred}] = \Pi \cup \Gamma \cup \Xi_{\text{pre}}$
- $\text{tr}[\text{Args}_{P \cup H}, \text{attacks}_{P \cup H}; \text{grounded}] = \Pi \cup \Gamma \cup \Xi_{\text{gr}}$
- $\text{tr}[\text{Args}_{P \cup H}, \text{attacks}_{P \cup H}; \text{stable}] = \Pi \cup \Gamma \cup \Xi \cup \{\leftarrow \text{undec}(X)\}$

Π は以下の 1~7 のルールで構成される.

1. $\text{ag}(a) \leftarrow \text{hyp}(A).$
(但し $A \in \text{Args}_{P \cup H}$, $a = \text{name}(A)$, $\text{hyp}(A) \subseteq H$)
2. $\text{def}(a, b) \leftarrow \text{ag}(a), \text{ag}(b).$
(但し $(A, B) \in \text{attacks}_{P \cup H}$, $a = \text{name}(A)$, $b = \text{name}(B)$)
3. $\text{goal}(c) \leftarrow.$ (但し $C \in \text{Args}_{P \cup H} \setminus \text{Args}_H$, $c = \text{name}(C)$)
4. $\text{in}(X) \leftarrow \text{ag}(X), \text{not ng}(X).$
 $\text{ng}(X) \leftarrow \text{ag}(X), \text{ag}(Y), \text{in}(Y), \text{def}(Y, X).$
 $\text{ng}(X) \leftarrow \text{ag}(X), \text{ag}(Y), \text{undec}(Y), \text{def}(Y, X).$
5. $\text{out}(X) \leftarrow \text{ag}(X), \text{ag}(Y), \text{in}(Y), \text{def}(Y, X).$
6. $\text{undec}(X) \leftarrow \text{ag}(X), \text{not in}(X), \text{not out}(X).$
7. $\text{ok}(X) \leftarrow \text{ag}(X), \text{goal}(X).$
 $\leftarrow \text{not ok}.$

Γ は以下の 8~10 のルールで構成される.

8. $m_1(L_t) \leftarrow L$. $m_1(\delta_h) \leftarrow h$. (但し $L \in \mathcal{I}$, $h \in C_H$)
9. $m_2(\alpha, j) \leftarrow$. $m_2(L_t, j) \leftarrow$. $m_2(\delta_h, j) \leftarrow$. $eno(j) \leftarrow$.
(但し Π の任意の解集合 $S_j (1 \leq j \leq \xi)$ について, $L_t \in C'$ は
 $L \in S' \mid_{\mathcal{I}}$ を, $\delta_H \in C_H$ は $h \in S' \mid_H$ を表している.
 α は $goal(\alpha \in S_j)$)
10. $i(L_t) \leftarrow$. $ab(\delta_h) \leftarrow$.
但し $\mathcal{I} = \{in(a) \mid A \in \text{Args}_{P \cup H}, a = \text{name}(A)\}$
 $C = \{L_t \mid L_t \text{ はアトム } L \in \mathcal{I} \text{ を表す固体変数}\}$
 $C_H = \{\delta_h \mid \delta_h \text{ はリテラル } h \in H \text{ を表す固体変数}\}$

Ξ , Ξ_{nr} , Ξ_{gr} は以下の 11~14 のルールで構成される.

11. $m_1(X) \leftarrow goal(X), ag(X).$
 $f_1(X) \leftarrow cno(Y), m_1(X), goal(X), not\ m_2(X, Y).$
 $f_2(Y) \leftarrow goal(X), m_2(X, Y), not\ m_1(X).$
 $f(Y) \leftarrow cno(Y), not\ f_1(Y), f_2(Y).$
 $c_1(Y) \leftarrow f(Y), m_1(X), ab(X), not\ m_2(X, Y).$
 $d_1(Y) \leftarrow f(Y), m_2(X, Y), ab(X), not\ m_1(X).$
 $\leftarrow c_1(Y), not\ d_1(Y).$
12. $h_1(Y) \leftarrow cno(Y), m_1(X), ab(X), not\ m_2(X, Y).$
 $h_2(Y) \leftarrow ab(X), m_2(X, Y), not\ m_1(X).$
 $h(Y) \leftarrow cno(Y), not\ h_1(Y), not\ h_2(Y).$
 $c(Y) \leftarrow h(Y), m_1(X), i(X), not\ m_2(X, Y).$
 $d(Y) \leftarrow h(Y), m_2(X, Y), i(X), not\ m_1(X).$
13. $\leftarrow d(Y), not\ c(Y).$
14. $\leftarrow c(Y), not\ d(Y).$

但し $\Xi: 11$ $\Xi_{pr}: 11, 12, 13$ $\Xi_{gr}: 11, 12, 14$

定理 1 (発想的議論における説明 [5]) $\langle P, H \rangle$ と基礎リ
テラル $G \not\models H$ が与えられた時, $E \subseteq H$ が $AF_{P \cup E}$ に
おいて G を *claim* に持つ論証 $Ag \in \text{Args}_{P \cup H}$ につい
ての *skeptical* (または *credulous*) な説明であるのは, 無矛
盾である $\langle \text{tr}[\text{Args}_{P \cup H}, \text{attacks}_{P \cup H}; S_{\text{name}}], H \rangle$ に関
して $E \subseteq H$ が $a = \text{name}(Ag)$, $G = \text{claim}(Ag)$ である
ような観測 $\text{in}(a)$ についての *skeptical* (または
credulous) な説明である場合であり, その時に限る.

文献 [5] では、極小の説明については言及していない。本研究で提案する、一貫性制約を用いて所与の説明集合より $\mathcal{C}$ に関する極小の説明を求める方法を以下に示す。

定義 12 (一貫性制約) 定理 1 で得た $E = \{h_i \in H \mid 1 \leq i \leq k\}$ に関して一貫性制約 $IC(E)$ は以下で定義される。

定理 2 (極小説明の計算) 説明 E_i が $S = \{E_i \mid E_i \subseteq H, 1 \leq i \leq n\}$ において $\subseteq$ に関して極小であるのは

$$\bigcup IC(E_i) \cup E_i \text{ が無矛盾であり, その時に限る.}$$
$$1 \leq j \leq n, \quad j \neq i$$

3.3 例題

議題” “プロジェクトは終了するか: *finish(project)*” が与えられた時、議題に対する *preferred semantics* における *skeptical* な説明を発想的議論によって求める。エージェントの知識は以下の $\langle PH \rangle$ で与えられる。

P	$\begin{aligned} &fin(pro) \leftarrow arr(p), \text{ pay}(m), \text{ not } \neg fin(pro). \\ &fin(pro) \leftarrow w(j). \\ &\neg fin(pro) \leftarrow \text{not } w(j). \\ &w(j) \leftarrow w(ma). \\ &\neg w(j) \leftarrow ho(pro). \end{aligned}$
H	$\{ arr(p), \text{ pay}(m), ho(pro), w(ma) \}$

$Args_{P \cup H}$ は以下の論証 $a1 \sim a9$ からなる.

- $a1 = [fin(pro) \leftarrow arr(p), pay(m), not \neg fin(pro);$
 $\quad arr(p) \leftarrow; pay(m) \leftarrow]$
 $a2 = [fin(pro) \leftarrow w(j); w(j) \leftarrow w(ma); w(ma) \leftarrow]$
 $a3 = [\neg fin(pro) \leftarrow not w(j)]$
 $a4 = [w(j) \leftarrow w(ma); w(ma) \leftarrow]$
 $a5 = [\neg w(j) \leftarrow h(pro); h(pro) \leftarrow]$
 $a6 \sim a9$ は H の各要素を頭部に持つ本体の無いルール
 1つからなる論証である

議題を結論に持つ論証は a_1 と a_2 の二つ存在する。
 AAF_{PUE} (但し $E \subseteq H$) に *reinstatement labelling*[3]

を付与したものを図1に示す。 $tr[Arg_{SPUH}, attacks_{SPUH}; preferred] \cup \Gamma(H)$ は図1の太字のラベリングを表す解集合を生成する。 $tr[Arg_{SPUH}, attacks_{SPUH}; preferred] \cup \Gamma(H) \cup \{t \leftarrow E_6\} \cup \{\leftarrow not\ t\} \cup \{\leftarrow in(a1)\}$ が矛盾となることから $a1$ に対する *skeptical* な説明 E_6 が得られる。同様に $a2$ に対する *skeptical* な説明は E_3, E_6 となる (但し $E_3 \subseteq E_6$)。ここで $\Gamma(H) = \{h' \leftarrow not\ h, h \leftarrow not\ h' \mid h \in H\}$ とする。また $E_3 \cup IC(E_6)$ が無矛盾であるので E_3, E_6 の内極小説明は E_3 となる。よって $a1$ ($a2$) に関する極小説明は E_6 (E_3) となり、議題を導く全て説明の中での極小説明は $a2$ に関する E_3 となる。

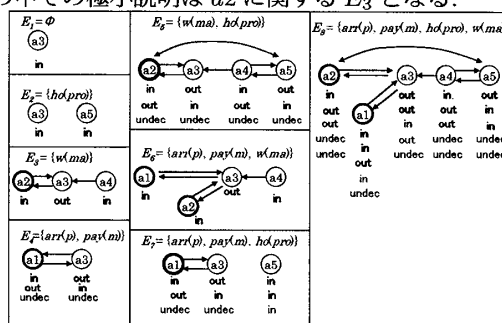


図 1: $AAF_{P||E}(a6 \sim a9$ は省略)

4 実装・評価

本研究の提案手法を ASP ソルバ: DLV と Java 言語を用いて実装した。発想的プログラムを記述したファイルを入力とし、議題、意味論、質問応答、攻撃の概念を指定し議題に対する説明を計算し出力する。論證ごとの極小説明が欲しい時は `minimal`、全説明の中の極小説明が欲しい時は `minimalAll` を指定する。極小化の指定があった場合は発想的議論によって求めた説明を入力とし、定理 2 に従って極小の説明を計算し出力する。例題を用いた実行例を以下の図 2 に示す。これより議題に対する説明・極小説明が正しく計算されていることがわかる。

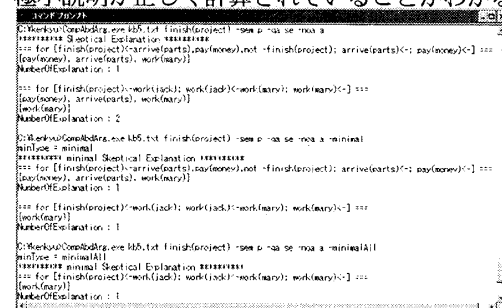


図 2: 例題の実行結果

5 おわりに

本研究では発想的議論の計算を行うツールの実装を行った。様々な例題を用いてツールの動作検証を行い、正しい計算結果を得られることを確認した。また本研究は文献 [6] のエージェント間交渉の研究にも用いられており有用性が検証された。

参考文献

- [1] 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.
- [2] R. Schweimeier, M. Schroeder: A Parameterised Hierarchy of Argumentation Semantics for Extended Logic Programming and its Application to the Well-founded Semantics, *Theory and Practice of Logic Programming*, pp.207-242, 2005.
- [3] M.Caminada: On the issue of reinstatement in argumentation, *Proc. of JELIA-2006, LNAI, Vol.4160*, pp.111-123. 2006
- [4] Skama, C., Inoue, K.: An abductive framework for computing knowledge base updates. *TPLP* 3(6): 671-713 (2003)
- [5] T.Wakaki K.Nitta H.Sawamura: Computing Abductive Arguments in Answer Set Programming, *Proc. of ARGMAST 2009*, pp. 189-206, 2009
- [6] 関口知之：発想的議論を用いた譲歩と妥協の形式化に基づくエージェント間交渉, 情報処理学会第 72 回全国大会 2010 年 3 月発表予定