解集合プログラミングによる議論の意味論の計算

伊藤 浩太[1] 若木 利子[2]
芝浦工業大学 システム工学部*

1 はじめに

複数人間が議論に基づき種々の問題解決をするプロセスをマルチエージェントにシミュレートさせるための形式化として，1995年，P.M. Dungによって議論の枠組の4種類の意味論：complete/preferred/grounded/stable semanticsが提案された[1]．近年，このような背景の下に，各意味論とそれに関する質問応答の種々の計算手続きが提案されている．他方，人間の行う議論は本質的に非単調推論である．LPNMRの分野では解集合プログラミング(ASP)が非単調推論を必要とする問題に有効な問題解決パラダイムとして世界的に認知され，これまでにASPプログラムを計算するASPソルバが数多く開発されている．しかし，議論の推論計算にASPを適用する試みは殆ど存在しない．そこで本研究では，2006年，Caminadaにより提案された議論の枠組でのreinstatement labellings[2]の手法と極小限定をASPで計算する著者らの方法[4]に基づき，議論の意味論とそれに関する質問応答を解集合プログラミングを用いて計算する方法を提案する．また，提案手法に基づいて各意味論や質問応答を自動で計算するプログラムをASPソルバのDLV[5]とC言語を用いて実装し，その検証，評価を行った．

2 準備

2.1 Dung's Standard Semantics

Dungが定義した議論の意味論を以下に示す[1]．

定義1 (議論フレームワーク) 議論フレームワークは$(Ar,\ def)$で表現される．但し，Arは論証集合，defはAr上の2項関係(*i.e.* $def \subseteq Ar \times Ar$)である．$(a,\ b) \in def$とは，"$a$が$b$を攻撃する"と言う意味を表す．

定義2 ($defend$ / $conflict\ free$) Arを論証集合，SをArの部分集合とする．この時，$(a,\ b) \in def$なる論証$a,\ b \in S$が存在しなければ，"集合Sは$conflict\ free$である"と言う．また，論証$a \in Ar$について，$(b,\ a) \in def$なる論証$b \in Ar$が存在する時，$(c,\ b) \in def$なる論証$c \in S$が存在すれば，"aはSによって$defend$される"と言う．

定義3 (議論の意味論) $Args \subseteq Ar$を$conflict\ free$な論証集合，単調関数$F : 2^{Ar} \rightarrow 2^{Ar}$を$F(Args) = \{A \mid A$は$Args$により$defend$される$\}$と定義する．この時，議論の各意味論($semantics$)は以下で定義される各$extension$で与えられる．

- $Args$が$complete\ extension$ iff $Args = F(Args)$.
- $Args$が$preferred\ extension$
  iff $Args$は$\subseteq$に関する極大な$complete\ extension$.
- $Args$が$grounded\ extension$
  iff $Args$は$\subseteq$に関する極小な$complete\ extension$.
- $Args$が$stable\ extension$
  iff $preferred\ extension$なる$Args$について，$Ar \setminus Args$の任意の論証を攻撃する論証が$Args$に存在．

Computing Argument-based Extensions in Answer Set Programming
[1]Kota Ito,[2]Toshiko Wakaki
*Shibaura Institute of Technology

2.2 Reinstatement Labellings

文献[2, 3]のreinstatement labellingsを以下に示す．

定義4 (AF-$labelling$) 議論フレームワーク$(Ar,\ def)$のAF-labellingは(全)関数$\mathcal{L} : Ar \rightarrow \{in,\ out,\ undec\}$である．これに対して$in(\mathcal{L})$, $out(\mathcal{L})$, $undec(\mathcal{L})$を次のように定義する．

- $in(\mathcal{L}) \stackrel{\text{def}}{=} \{a \in Ar \mid \mathcal{L}(a) = in\}$
- $out(\mathcal{L}) \stackrel{\text{def}}{=} \{a \in Ar \mid \mathcal{L}(a) = out\}$
- $undec(\mathcal{L}) \stackrel{\text{def}}{=} \{a \in Ar \mid \mathcal{L}(a) = undec\}$

定義5 ($Reinstatement\ Labellings$) AF-labelling:$\mathcal{L}$が次の条件を満たす時，reinstatement labellingと言う．

- $\forall a \in Ar : (\mathcal{L}(a) = out \equiv \exists b \in Ar : (b\ def\ a \wedge \mathcal{L}(b) = in))$
- $\forall a \in Ar : (\mathcal{L}(a) = in \equiv \forall b \in Ar : (b\ def\ a \supset \mathcal{L}(b) = out))$

$(Ar,\ def)$の任意のreinstatement labelling $\mathcal{L}$と，各意味論の$extension$ Eについて，以下の結果が得られる．

- $complete\ extension$ $E = \{a \in Ar \mid \mathcal{L}(a) = in\}$
- $preferred(resp.\ grounded)\ extension$
  $E = \{a \in Ar \mid a \in in(\mathcal{L})$ 但し，$\mathcal{L}$は$in(\mathcal{L})$が極大$(resp.$ 極小$)$となるreinstatement labelling$\}$
- $stable(resp.\ semi\text{-}stable)\ extension$
  $E = \{a \in Ar \mid a \in in(\mathcal{L})$ 但し，$\mathcal{L}$は$undec(\mathcal{L}) = \phi(resp.\ undec(\mathcal{L})$が極小$)$となるreinstatement labelling$\}$

例1. 次のような$(Ar,\ def)$が与えられているとする．
但し，Ar=$\{a,\ b,\ c\}$, def=$\{(a,\ b),\ (b,\ a),\ (b,\ c)\}$
これのreinst. labellingsは図1の3通りが存在する．

$\mathcal{L}1$	in	out	in
$\mathcal{L}2$	out	in	out
$\mathcal{L}3$	$undec$	$undec$	$undec$

図1: $(Ar,\ def)$のreinst. labellings

よって，complete extensionは$\{a,\ c\}$, $\{b\}$, $\{\}$の3個，preferred extensionは$\{a,\ c\}$, $\{b\}$の2個，grounded extensionは$\{\}$の1個となる．

定義6 ($Sceptical\ and\ Credulous\ Query\text{-}Answering$) 所与の議論フレームワーク$(Ar,\ def)$について，

- 論証$a \in Ar$が所与の意味論の下で$sceptical$ $(resp.\ credulous)$に正当化される
  iff 所与の意味論の全ての$(resp.$ ある$)\mathcal{L}$の集合のラベリングにおいて，$\mathcal{L}(a) = in$なる$a \in Ar$が存在．
- 論証$a \in Ar$が所与の意味論の下で$sceptical$ $(resp.\ credulous)$に却下される
  iff 所与の意味論の全ての$(resp.$ ある$)\mathcal{L}$の集合のラベリングにおいて，$\mathcal{L}(a) = out$なる$a \in Ar$が存在．
- それ以外の場合，論証$a \in Ar$は所与の意味論の下で$sceptical(resp.\ credulous)$に$defensible$.

2.3 解集合プログラミング

定義7 本研究で扱うASPプログラムは，以下のルールの集合のNLP(一般論理プログラム)である．

$$A \leftarrow B_1, \cdots, B_m, not\ B_{m+1}, \cdots, not\ B_n$$
$$\leftarrow B_1, \cdots, B_m, not\ B_{m+1}, \cdots, not\ B_n$$

ここで，$n \geq m \geq 0$, A, B_iは原子式である．notはNAFオペレータである．NLPの宣言的意味は安定モデル(即ち，解集合)で与えられる[4]．

3 ASPによる議論の意味論の計算

本研究の提案手法では，議論フレームワークより各意味論に対応した変換論理プログラムを構成し，その解集合を計算することで意味論の *extensions* を求める．

3.1 変換論理プログラムの構成

定義 8 所与の議論フレームワーク $(Ar,\ def)$ より，NLP $\Pi \stackrel{\text{def}}{=} \Pi_{AF} \cup \Pi_{Lab}$ を構成する．但し，Π_{AF} は 1〜2，Π_{Lab} は 3〜5 のルールからなる．

1. $ag(a) \leftarrow$ *for any* $a \in Ar$
2. $def(a,\ b) \leftarrow$ *for any* $(a,\ b) \in def$
3. $in(X) \leftarrow ag(X),\ not\ ng(X)$
   $ng(X) \leftarrow in(Y),\ def(Y,\ X)$
   $ng(X) \leftarrow undec(Y),\ def(Y,\ X)$
4. $out(X) \leftarrow in(Y),\ def(Y,\ X)$
5. $undec(X) \leftarrow ag(X),\ not\ in(X),\ not\ out(X)$

定義 9 所与の議論フレームワーク $(Ar,\ def)$ より，*complete/stable* の意味論に対応した次の NLP の変換論理プログラムを構成する．

- $tr[Ar,\ def;\ \text{"complete"}] \stackrel{\text{def}}{=} \Pi$
- $tr[Ar,\ def;\ \text{"stable"}] \stackrel{\text{def}}{=} \Pi \cup \{\leftarrow undec(X)\}$

定義 10 S を解集合，X を集合とする．この時，$S|_X \stackrel{\text{def}}{=} S \cap X$ を"S の X-射影"と呼ぶ．

定義 11 $AS = \{S \mid S$ は Π の解集合 $\}$，$\xi : |AS|$，ψ は $\psi : AS \rightarrow \{1, \cdots, \xi\}$ なる全単射関数とする．$\psi(S) = j$ なる解集合 $S \in AS$ が存在する時，"j" を S の *cardinal number* と呼ぶ．

定義 12 議論フレームワーク $(Ar,\ def)$ に対して，$\mathcal{I}$, $\mathcal{U}$ を次のような原子式の集合として定義する．

$\mathcal{I} = \{in(a) \mid a \in Ar\}$ $\mathcal{U} = \{undec(a) \mid a \in Ar\}$

定義 13 $\mathcal{C}$ を次のような定数の集合として定義する．

$\mathcal{C} = \{L_t \mid L_t$ は $L \in \mathcal{I} \cup \mathcal{U}$ を表す個体定数 $\}$

定義 14 所与の議論フレームワーク $(Ar,\ def)$ より，*preferred/grounded/semistable* の意味論に対応した次の NLP の変換論理プログラムを構成する．

- $tr[Ar,\ def;\ \text{"preferred"}] \stackrel{\text{def}}{=} \Pi \cup \Gamma \cup \Xi_{pr}$
- $tr[Ar,\ def;\ \text{"grounded"}] \stackrel{\text{def}}{=} \Pi \cup \Gamma \cup \Xi_{gr}$
- $tr[Ar,\ def;\ \text{"semistable"}] \stackrel{\text{def}}{=} \Pi \cup \Gamma \cup \Xi_{semi}$

ここで，Γ は次の 1〜3.2 の *domain-dependent* なルールで構成される．

1. $m_1(L_t) \leftarrow L$ *for any* $L \in \mathcal{I} \cup \mathcal{U}$
   $L_t \in C$ は L を表す個体定数
2. Π の解集合 S の $\mathcal{I} \cup \mathcal{U}$-射影 $S|_{\mathcal{I} \cup \mathcal{U}}$ に対して，
   $m_2(L_t,\ j) \leftarrow$, $cno(j) \leftarrow$ $(1 \leq j \leq \xi)$
   $L_t \in C$ は $L \in S_j|_{\mathcal{I} \cup \mathcal{U}}$ を表す個体定数

3.1 $i(L_t) \leftarrow$ *for any* $L \in \mathcal{I}$
   $L_t \in C$ は L を表す個体定数

3.2 $u(L_t) \leftarrow$ *for any* $L \in \mathcal{U}$
   $L_t \in C$ は L を表す個体定数

また，Ξ_{pr} は次の 4 と 6，Ξ_{gr} は 4 と 7，Ξ_{semi} は 5 と 7 の *domain-independent* なルールで構成される．

4. $c(Y) \leftarrow cno(Y),\ m_1(X),\ i(X),\ not\ m_2(X,\ Y)$
   $d(Y) \leftarrow m_2(X,\ Y),\ i(X),\ not\ m_1(X)$
5. $c(Y) \leftarrow cno(Y),\ m_1(X),\ u(X),\ not\ m_2(X,\ Y)$
   $d(Y) \leftarrow m_2(X,\ Y),\ u(X),\ not\ m_1(X)$
6. $\leftarrow d(Y),\ not\ c(Y)$
7. $\leftarrow c(Y),\ not\ d(Y)$

3.2 健全性/完全性定理と質問応答

所与の議論フレームワーク $(Ar,\ def)$ が与えられており，*Sname* は意味論名の 1 つを表すとする．

定理 1 (健全性/完全性定理)
E が意味論名 *Sname* の *extension* であるならば，NLP $tr[Ar,\ def;\ Sname]$ は $M|_{\mathcal{I}} = \{in(a) \mid a \in E\}$ なる解集合 M を持つ．逆に，NLP $tr[Ar,\ def;\ Sname]$ が $M|_{\mathcal{I}} = \{in(a) \mid a \in Ar\}$ なる解集合 M を持つならば，$E = \{a \mid in(a) \in M|_{\mathcal{I}}\}$ なる議論フレームワーク $(Ar,\ def)$ の意味論 *Sname* の *extension* E を持つ．

定理 2 (質問応答)

- $tr[Ar,\ def;\ Sname] \cup \{\leftarrow in(a)\}$ が矛盾
  iff $a \in Ar$ は *Sname* の下で *sceptical* に正当化．
- $tr[Ar,\ def;\ Sname] \cup \{\leftarrow not\ in(a)\}$ が無矛盾
  iff $a \in Ar$ は *Sname* の下で *credulous* に正当化．
- $tr[Ar,\ def;\ Sname] \cup \{\leftarrow out(a)\}$ が矛盾
  iff $a \in Ar$ は *Sname* の下で *sceptical* に却下．
- $tr[Ar,\ def;\ Sname] \cup \{\leftarrow not\ out(a)\}$ が無矛盾
  iff $a \in Ar$ は *Sname* の下で *credulous* に却下．
- それ以外の場合，$a \in Ar$ は *Sname* の下で *sceptical(resp. credulous)* に *defensible*.

4 実装・検証・評価

本研究の提案手法を ASP ソルバの DLV と C 言語を用いて実装した．議論フレームワークを記述したファイルを入力とし，任意の意味論と質問応答の種類，及び質問応答をする論証を実行オプションとして指定し，その計算結果を出力する．種々の例を用いて検証を行った結果，正しく計算されていることを確認した．例 1 の実行例を図 2 に示す．

また，*preferred/grounded extensions* を計算する際に，*in* のラベルについての極大・極小を判定するほかに，*out* のラベルについて同様の判定をして，先の *extensions* を求めることができる [2]．*out* のラベルについての極大・極小を判定するプログラムを作成し，同様の検証を行った結果，*in* のラベルによる判定のものと一致する結果を得ることができた．

```
C:¥workspace>Comp_Arg_in.exe ex1.txt -complete -preferred -grounded
**********Complete Extension**********
{a, c}
{b}
{}

**********Preferred Extension**********
{b}
{a, c}

**********Grounded Extension**********
{}
```

図 2: 例 1 の実行例

5 おわりに

本研究では議論フレームワークの 5 種類の意味論を reinstatement labellings の手法を用いて ASP で計算する方法を提案し，ASP ソルバの DLV と C 言語を用いて実装し，その検証，評価を行った．そして，種々の例を用いて検証を行い，正しい計算結果を得ることができた．また，本研究で作成した議論計算プログラムは Linux と Windows 上で稼動しており，近く Web 上に公開予定である．

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] M. Caminada: On the issue of reinstatement in argumentation, Proc. of 10th European Conference on Logics in Artificial Intelligence (JELIA06), pp. 111-123, LNAI No.4160, Springer, 2006.

[3] M. Caminada: Semi-stable semantics, In P.E. Dunne and T.J.M. Bench-Capon, editors, Computational Models of Argument; Proc. of COMMA 2006, pp. 121-130, IOS Press, 2006.

[4] 若木利子，富田一夫：生成と検査の論理プログラムの統合による極小限定・定理証明器の構築，人工知能学会論文誌，Vol22, No.5, pp.472-481, 2007.

[5] dlv URL http://www.dbai.tuwien.ac.at/proj/dlv/