2D-4

# 動的プリファレンスを扱う論理プログラムの解集合プログラミングによる推論計算

村山 知明[*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[*3]

## 1　はじめに

日常生活でコンフリクトする結論が導かれた時, 人間は"課長の意見より部長の意見を優先する"などのプリファレンス (選好, 優先度) の知識を用いてコンフリクトを解消し, 推論や意思決定を行っている. プリファレンスには外から与えられてコンフリクトの解消に用いられる静的プリファレンスの他, 法廷や司法の場でしばしば"新法優先"や"上位法優先"という法令間のプリファレンス, 更に"新法優先より上位法優先を優先する"というメタなプリファレンスが用いられている. この結果, プリファレンスも推論の対象とする動的プリファレンスを扱う問題となる. しかし, 坂間・井上の PLP[3] は静的プリファレンスを扱っており, 動的プリファレンスは表現できない. そこで動的プリファレンスも扱えるように表現力を高めた階層的 PLP[4] が昨年, 若木により提案された. PLP 及び階層的 PLP の意味論はそれぞれ p-answer sets で与えられる. 2004 年黄檗が開発した"PLP の優先的解集合計算プログラム"[1] においては, PLP の p-answer set を複数の ASP プログラムの実行により計算していた. 本研究では動的プリファレンス (静的プリファレンス) を扱う階層的 PLP (PLP) の p-answer set を 1 つの解集合から求める ASP プログラムを提案し, それに基づく推論エンジンを開発した.

## 2　準備

### 2.1　解集合プログラミング

解集合プログラミング (以後 ASP) とは論理プログラムを用いて問題の知識を表現し, その問題の解を解集合から得るプログラミング手法である.

### 2.2　PLP と p-answer set

優先度付き論理プログラム (*Prioritized logic program*, 以後 PLP) の構文とその意味論を以下に示す.

**定義 1（拡張論理プログラム [2]）** 拡張論理プログラム (以後 ELP) とは次のような形をしたルールの集合である.

$$L \leftarrow L_1, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n$$

ただし $L$ はリテラル (原子式 $A$ またはその論理否定 $\neg A$), $not$ は NAF(Negation as Failure) 否定である.

**定義 2（解集合 [2]）** $Lit_P$ を $P$ の言語の基礎リテラルの集合"と定義する. ELP $P$ の解集合 $S$ は $Lit_P$ の部分集合である (解集合の定義の詳細は [2] 参照). $P \models L$ は $P$ の任意の解集合 $S$ に対して, $L \in S$ を意味する.

**定義 3（優先度付き論理プログラム [3]）** リテラル $e, e'$ 間の優先度を $e \preceq e'$ で表す. $\preceq$ は反射律, 推移律を満たす擬順序 (preorder) である. PLP は $(P, \Phi)$ で定義される. $P$ は ELP, $\Phi$ は優先度の集合で, $\Phi \subseteq Lit_P^2$ である.

PLP $(P, \Phi)$ の意味論は, $P$ の解集合の中で $\Phi$ の優先情報に基づいて最も望ましい preferred answer set (以後 p-answer set)[3] の集合により与えられる.

Computing preferred answer sets of hieralcal PLP
*1 Tomoaki Murayama
*2 Toshiko Wakaki
*3 Shibaura Institute of Technology

## 3　階層的 PLP と p-answer sets

階層的 PLP[4] の構文と意味論を以下に示す. 階層的 PLP の意味論は p-answer sets で与えられる. 以下で $N$ をルール名の集合とする. $\preceq (n_{r1}, n_{r2}, n_\delta)$ は priority atom と称され, 「ルール名 $n_\delta \in N$ を持つルール $\delta$ により, "ルール名 $n_{r2} \in N$ のルール $r2$ がルール名 $n_{r1} \in N$ のルール $r1$ より高い優先度を持つ"」ことを意味する. なお $\delta$ は, $r1, r2$ のルールより 1 レベル上位のメタなルールであり, $r1, r2$ は同一メタレベルのルールである.

**定義 4（階層的 PLP[4]）** 階層的 PLP は $(P, \Psi, \Phi)$ で定義される. $P$ は ELP で, 以下の (1) の形式の名前のないルール, または (2) の形式の名前 $n_r \in N_0$ のついたルール $r$ の集合である.(但し, $N_0 \in N$)

$$L_0 \leftarrow L_1, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n. \quad (1)$$

$$n_r: \quad L_0 \leftarrow L_1, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n. \quad (2)$$

$\Psi$ は ELP で, priority atom $\preceq (n_{r1}, n_{r2}, n_\delta)$ を head に持つ以下のの形のルール $\delta$ の集合である.

$$n_\delta: \preceq (n_{r1}, n_{r2}, n_\delta) \leftarrow L_1, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n.$$

$\Phi$ は以下のリテラル集合 $\mathcal{L}$ 上の関係 $\preceq$ で定義される優先度の集合である.

$$\mathcal{L} \stackrel{\text{def}}{=} Lit_P \cup N_0 \setminus \{L | L = head(r)\ for\ r \in P\ named\ n_r\}$$

名前 $n_r, n_{r'} \in N_0$ を持つルール $r, r' \in P$ について,

$$r \preceq r' \text{ iff } head(r) \preceq head(r') \text{ iff } \preceq (n_r, n_{r'}, n_\delta)$$

が成り立つ. ここで $r \preceq r'$ は「ルール $r$ よりルール $r'$ を優先する」ことを意味する. $\Phi$ の定義は, このようなルール $r, r'$ 間の優先度は $head(r) \preceq head(r') \in \Phi$ ではなく, $\preceq (n_r, n_{r'}, n_\delta) \leftarrow \in \Psi$ で与えることを意味する.

次に階層的 PLP $(P, \Psi, \Phi)$ の意味論を定義する.

**定義 5** NLP $\Pi$ と $\Delta$[4] を以下に示す. 名前 $n_r \in N$ を持つ $r \in P \cup \Psi$ に対して $\Delta$ のルールが定義される.

$$\begin{aligned}
\Pi: & \prec (X, Y) \leftarrow \preceq (X, Y, Z), \preceq (Y, X, U), X \neq Y \\
& \qquad\qquad not \prec (Y, X), not \prec (Z, U). \\
& \prec (X, Y) \leftarrow \preceq (X, Y, Z), not\ ng(Y, X). \\
& ng(Y, X) \leftarrow \prec (Y, X, U). \\
& \preceq (X, Y, t(U, V)) \leftarrow \preceq (X, Y, U), \preceq (Y, Z, V). \\
\Delta: & n_r \leftarrow body(r).
\end{aligned}$$

**定義 6** $\Phi_d$ と $\Phi_s$ を以下で定義する [4].

$$\Phi_d \stackrel{\text{def}}{=} \{(n, n') \mid P \cup \Psi \cup \Pi \cup \Delta \models \prec (n, n')\}$$

$$\Phi_s \stackrel{\text{def}}{=} \{(e, e') \mid (e, e') \in \Phi^* \wedge (e', e) \notin \Phi^*\}$$

ここで $\Phi^*$ は $\Phi$ の推移閉包を表す.

**定義 7（解集合間のプリファレンス $\sqsubseteq_{as}$[4]）** 所 与 の PLP $(P,\ \Psi,\ \Phi)$ に対して $(\Phi_d \cup \Phi_s)^*$ を $\prec$, $P \cup \Psi \cup \Pi \cup \Delta$ の解集合の集合を $\mathcal{AS}$ と定義する. $\mathcal{AS}$ 上の関係 $\sqsubseteq_{as}$ は以下の (i)∼ (iii) で定義される. 任意の $M_1, M_2, M_3 \in \mathcal{AS}$ に対して

(i)　$M_1 \sqsubseteq_{as} M_1$

(ii)　$M_1 \sqsubseteq_{as} M_2$ であるとは, 以下の条件 (a), (b) を満たすあるリテラル $e_2 \in M_2 \setminus M_1$ が存在することである.

　(a)　$(e_1 \prec e_2) \in (\Phi_d \cup \Phi_s)^*$ なる $e_1 \in M_1 \setminus M_2$ が存在し,

　(b)　$(e_2 \prec e_3) \in (\Phi_d \cup \Phi_s)^*$ なる $e_3 \in M_1 \setminus M_2$ が存在しない.

(iii)　$M_1 \sqsubseteq_{as} M_2$ かつ $M_2 \sqsubseteq_{as} M_3$ であるならば $M_1 \sqsubseteq_{as} M_3$

他方, $M_1 \sqsubset M_2 \stackrel{\text{def}}{=} (M_1 \sqsubseteq M_2) \wedge (M_2 \not\sqsubseteq M_1)$ である.

**定義 8（p-answer set[4]）** $(P, \Psi, \Phi)$ を階層的 PLP, $\mathcal{AS}$ を $P \cup \Psi \cup \Pi \cup \Delta$ の解集合の集合とする. このとき, 解集合 $M \in \mathcal{AS}$ が任意の解集合 $M' \in AS$ に対して, $M \not\sqsubset_{as} M'$ のとき, $S = M|_{Lit_P}$ なる $P$ の解集合 $S$ を PLP $(P, \Psi, \Phi)$ の p-answer set と定義する.

## 4　ASP による階層的 PLP の p-answer sets の計算

本研究では, 階層的 $\mathrm{PLP}(P,\Psi,\Phi)$ の p-answer set と解集合が一対一に対応する, ELP のクラスの変換論理プログラム $T[P,\Psi,\Phi]$ を提案する. 集合 $\mathcal{C}$ を以下で定義する.

$\mathcal{C} \stackrel{\mathrm{def}}{=} \{L_t \mid L_t$はリテラル $L \in Lit_P$を表す個体定数$\}$

さらに ELP $P$, $\pi$, $T[P,\Psi,\Phi]$ の解集合を $S$, $M$, $\mathcal{M}$ で表すものとする. $\pi$ の解集合全体の集合 $\mathcal{AS}$ について, $M_j \in \mathcal{AS}$ $(1 \le j \le \xi)$ を $\pi$ の $j$ 番目の解集合を表すとする. (但し $\xi = |\mathcal{AS}|$) 本研究では $M_j \in \mathcal{AS}$ の情報を, 下記意味の述語記号 $m_1$, $m_2$ を用いた $\Gamma$ の no.1〜no.5 のルールとして変換論理プログラムの解集合 $\mathcal{M}$ に埋め込む. 以下でリテラル $L \in Lit_P$ を表す個体定数を $L_t \in \mathcal{C}$ で表す. iff は "if and only if" を意味する.

- $m1(L_t) \in \mathcal{M}$ iff $L \in S = M|_{Lit_P} = \mathcal{M}|_{Lit_P}$
- $L \in Lit_P$ に対して, $m2(L_t,j) \in \mathcal{M}$ iff $L \in M_j$
- $n \in N$ に対して, $m2(n,j) \in \mathcal{M}$ iff $n \in M_j$
- $m3(n_1,n_2,j) \in \mathcal{M}$ iff $\prec(n_1,n_2) \in M_j$

$(e_1,e_2) \in \Phi$ の情報は $\ll$ の述語記号を用いた $\Gamma$ の no.6 のルールで表現して $\mathcal{M}$ に埋め込む.

**定義 9（変換論理プログラム）** 所与の階層的 $\mathrm{PLP}(P,\Psi,\Phi)$ について, $\pi = P \cup \Psi \cup \Pi \cup \Delta$ とする. 変換論理プログラム $T[P,\Psi,\Phi]$ は以下の ELP で定義される.

$$T[P,\Psi,\Phi] \stackrel{\mathrm{def}}{=} \pi \cup \Gamma \cup \Xi$$

ここで, $\Gamma, \Xi$ は以下のルール集合である.

$\Gamma$ :

1. $m1(L_t) \leftarrow L.$ 　 *for each* $L \in Lit_P$. 但し,$L_t \in C$ は $L$ を表す定数.
2. $m1(n) \leftarrow n.$ 　 *for a head* $n$ *of a rule in* $\Delta$.
3. $m2(L_t,j) \leftarrow .$ 　 *for any* $L \in M_j|_{Lit_P}$. $L_t \in C$ は $L$ を表す定数.
4. $m2(n,j) \leftarrow .$ 　 *for* $n \in M_j|_N$
5. $m3(n_1,n_2,j) \leftarrow .$ *for* $\prec(n_1,n_2) \in M_j$
6. $\ll(L_{t_1},L_{t_2}) \leftarrow .$ *for* $(L_1,L_2) \in \Phi$.
$L_{t_1},L_{t_2} \in C$ は $L_1,L_2$ を表す定数.
ただし $L_1 \in N$(または $L_2 \in N$) であるならば
$\ll(L_1,L_{t_2}) \leftarrow .$(または $\ll(L_{t_1},L_2) \leftarrow .$) とする.

$\Xi$ :

7. $\ll(X,Z) \leftarrow \ll(X,Y), \ll(Y,Z)$
8. $<(X,Y) \leftarrow \ll(X,Y), not \ll(Y,X)$
9. $asno(N) \leftarrow m2(X,N).$
10. $\prec_{crd}(X,Y) \leftarrow \prec(X,Y), asno(N), not\ m3(X,Y,N).$
11. $<(X,Y) \leftarrow \prec(X,Y), not \prec_{crd}(X,Y).$
12. $<(X,Z) \leftarrow <(X,Y), <(Y,Z).$
13. $gr1(N,Y) \leftarrow m1(Y), m2(X,N), <(X,Y),$
$not\ m1(X), not\ m2(Y,N).$
14. $gr2(N,Y) \leftarrow m1(Y), m2(Z,N), <(Y,Z),$
$not\ m1(Z), not\ m2(Y,N).$
15. $\sqsupseteq(N) \leftarrow gr1(N,Y), not\ gr2(N,Y).$
16. $rgr1(N,Y) \leftarrow m1(X), m2(Y,N), <(X,Y),$
$not\ m1(Y), not\ m2(X,N).$
17. $rgr2(N,Y) \leftarrow m1(Z), m2(Y,N), <(Y,Z),$
$not\ m1(Y), not\ m2(Z,N).$
18. $\sqsubseteq(N) \leftarrow rgr1(N,Y), not\ rgr2(N,Y).$
19. $grt1(N1,N2,Y) \leftarrow asno(N1), asno(N2), m2(X,N1),$
$m2(Y,N2), <(X,Y), not\ m2(Y,N1), not\ m2(X,N2).$
20. $grt2(N1,N2,Y) \leftarrow asno(N1), asno(N2), m2(Z,N1),$
$m2(Y,N2), <(Y,Z), not\ m2(Y,N1), not\ m2(Z,N2).$
21. $\sqsubseteq 2(N1,N2) \leftarrow grt1(N1,N2,Y), not\ grt2(N1,N2,Y).$
22. $\sqsubseteq 2(N1,N3) \leftarrow \sqsubseteq 2(N1,N2), \sqsubseteq 2(N2,N3).$
23. $\sqsupseteq(N2) \leftarrow \sqsupseteq(N1), \sqsubseteq 2(N2,N1).$
24. $\sqsubseteq(N2) \leftarrow \sqsubseteq(N1), \sqsubseteq 2(N1,N2).$
25. $\sqsubset(N) \leftarrow \sqsubseteq(N), not \sqsupseteq(N).$
26. $\leftarrow \sqsubset(N).$

$T[P,\Psi,\Phi] = \pi \cup \Gamma \cup \Xi$ の解集合 $\mathcal{M}$ について, $\Gamma \cup \Xi \setminus \{no.21$ のルール $\}$ は層状プログラムであるので, $\mathcal{M}$ は $\pi$ のある解集合 $M$ を包含する (i.e. $M \subseteq \mathcal{M}$). 従って, $\mathcal{M}$ に対して $M = \mathcal{M}|_{Lit_\pi}$ なる $\pi$ の解集合 $M$, 及び $S = \mathcal{M}|_{Lit_P} = M|_{Lit_P}$ なる $P$ の解集合 $S$ が存在する. ルール 6, 7, 8 により, $<(n,n') \in \mathcal{M}$ ならば, $\Phi_s$ の情報が $<$ のアトムとして, $\mathcal{M}$ に埋め込まれる. ルール 9, 10 より, $\prec_{crd}(n,n') \in \mathcal{M}$ ならば, $m3(n,n',N) \notin \mathcal{M}$ (i.e. $\prec_{crd}(n,n') \notin M_N$) なる $M_N$ が存在する. つまり $\pi \not\models \prec_{crd}(n,n')$ を意味する. ルール 11 では $<(n,n') \in \mathcal{M}$ ならば, $\prec(n,n') \in \mathcal{M}|_{Lit_\pi}$ (i.e. $\prec(n,n') \in M$) かつ $\prec_{crd}(n,n') \notin M_N$ なる $M_N$ が存在しない, つまり $\pi \models \prec(n,n')$ を意味し, $(n,n') \in \Phi_d$ が計算される. ルール 12 はルール 8, 11 と共に, $<(n,n') \in \mathcal{M}$ ならば $(n,n') \in (\Phi_s \cup \Phi_d)^*$ を意味し, $(\Phi_s \cup \Phi_d)^*$ を計算する. ルール 15 より $\sqsupseteq(N) \in \mathcal{M}$ ならば, $M \sqsupseteq_{as} M_N$(定義 7(ii)), ルール 18 より $\sqsubseteq(N) \in \mathcal{M}$ ならば, $M \sqsubseteq_{as} M_N$(定義 7(ii)) が計算される. ルール 21 より $\sqsubseteq 2(N1,N2)$ ならば, $\mathcal{M}$ に埋め込まれた $M_{N1}$,$M_{N2}$ について, $M_{N1} \sqsubseteq_{as} M_{N2}$(定義 7(ii)) を計算する. ルール 22,23,24 は定義 7(iii) の推移律を計算する. ルール 26 は定義 8 に従い, $\mathcal{M}$ が表す $P$ の解集合 $\mathcal{M}|_{Lit_P}$ が p-answer set か否かを判定する.

**定理 1（健全性/完全性）** $S$ が $\mathrm{PLP}(P,\Psi,\Phi)$ の p-answer set ならば, $\mathcal{M}|_{Lit_P} = S$ なる $T[P,\Psi,\Phi]$ の解集合 $\mathcal{M}$ が存在する. 逆に $\mathcal{M}$ が $T[P,\Psi,\Phi]$ の解集合ならば, $\mathrm{PLP}(P,\Psi,\Phi)$ の p-answer set $S$ が存在する.

## 5　推論エンジンの実現と評価

階層的 $\mathrm{PLP}(P,\Psi,\Phi)$ の p-answer sets の計算, 及びそれらに関する質問評価を計算する推論エンジンを ASP ソルバ DLV と C 言語を用いて実装した. 以下に Gordon の法的推論の担保権の例題 [5] の実行を示す. 例題の詳細については紙面の都合上省略する. この例題を階層的 $\mathrm{PLP}(P,\Psi,\Phi)$ で記述すると以下のようになる.

```
 ucc : perf :- poss, not -perf.
sma : -perf :- ship, -file, not perf.
      poss.   ship.   -file.
      moreRecent(ucc, sma). fed(sma). state(ucc).
 lp1 : ⪯(sma,ucc,lp1) :- moreRecent(ucc,sma).
 ls1 : ⪯(ucc,sma,ls1) :- fed(sma), state(ucc).
 lex : ⪯(lp1,ls1,lex).
```

この例題を変換論理プログラム $T[P,\Psi,\Phi]$ に基づく推論エンジンで計算した結果を以下に示す. この例題では LS の原則が優先され, それにより法律 ucc が優先されるため, 担保権は完全でないとなる. 実行結果でも担保権は完全でない (*-perf* ) となっており, 人間の思考と一致する正しい結果が得られていることが分かる.

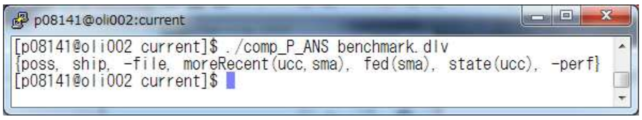


図 1　実行結果

また文献 [4] の example1,2 でも同様の検証を行い, 正しい結果が得られていることを確認した.

## 6　おわりに

本研究の成果は PLP の p-answer set を 1 つの解集合として求める ASP 記述の提案, 階層的 PLP の p-answer set を 1 つの解集合として求める ASP の提案, 及びその推論エンジンを開発である. なお, この変換プログラムは学内の HP にて公開予定である.

## 参考文献

[1] 黄檗 洋輔, 若木 利子: PLP の優先的解集合計算プログラムの開発と評価, 情報処理学会第 66 回全国大会講演論文集 (2), 3L-4, pp.31-32, 2004.3

[2] Gelfond, M. and Lifschitz, V. Classical Negation in Logic Programs and Disjunctive Databases. New Generation Computing 9, pp.365-385, 1991.

[3] C.Sakama and K.Inoue: Prioritized logic programming and its application to commonsense reasoning, Artificial Intelligence 123, pp.185-222, 2000

[4] T. Wakaki: Preference-Based Argumentation Handling Dynamic Preferences Built on Prioritized Logic Programming. Proc. of PRIMA-2011, LNSI vol. 7047, pp. 336-348, Springer (2011)

[5] Gordon, T.F., The pleadings game: An Artificial Intelligence Model of Procedural Justice, Ph.D. thesis, TU Darmstadt (1993)