

状態遷移図に基づく信念変更における修正と更新の使い分け

A Distinction Method of Revision and Update in Belief Change by State Transition Diagrams

工藤 康生*¹
Yasuo Kudo

村井 哲也*²
Tetsuya Murai

*¹室蘭工業大学工学部情報工学科

Department of Computer Science and Systems Engineering, Muroran Institute of Technology

*²北海道大学大学院情報科学研究科

Graduate School of Information Science and Technology, Hokkaido University

We propose a distinction method of revision and update of knowledge bases represented by extended logic programs. In order to accept new information into a knowledge base, we need to select either revising or updating the knowledge base. We use state transition diagrams to select either revision or update, which characterize revision of extended logic programs as syntactic minimal change, and update as semantic minimal change.

1. はじめに

信念変更 (belief change) は知識ベースに対して新たな情報を反映させる, 情報を取り除く等の変更操作を行うために, 情報の整合性を保ちつつ知識ベースを書き換える問題を扱う. 信念変更の分野において知識ベースに新たな情報を反映させる操作は, 修正 (revision) [Alchourrón 85, Katsuno 91] と更新 (update) [Katsuno 92] に大別される. 修正の操作は, 新情報が知識ベースと矛盾しない場合はそのまま追加し, 矛盾する場合は知識ベースに誤った情報が含まれていると見なして知識ベースを書き換える. これに対して更新の操作は, 新情報を状況の変化によるものとして捉え, 新情報により判明した現在の状況を知識ベースへ反映させるために知識ベースを書き換える. このように, 修正と更新は性質が異なる操作であるため, 新情報が得られた場合に, これら 2 種類の操作を使い分けられることが望ましい. しかしながら, 信念変更に関する従来研究では, 新情報に対する操作として修正と更新のどちらか一方のみを扱い, 両者の使い分けは考慮しないものが多い.

本稿では, 拡張論理プログラム (extended logic program) [Gelfond 91] として表現される知識ベースに対して新情報を反映させる場合に, プログラムの意味を規定する解集合 (answer set) を 1 個の状態と見なす状態遷移図を用いることで, 修正と更新を使い分ける手法を提案する.

2. 背景

2.1 知識ベースの修正及び更新

本稿では, Katsuno and Mendelzon [Katsuno 91] による知識ベースの修正を用いる. Katsuno and Mendelzon による修正の定式化では, 知識ベースを有限個の原始文から構成された古典命題論理の言語 \mathcal{L}_{CPL} の 1 個の論理式で表現する. 知識ベースを表す論理式 ψ を, 新情報を表す論理式 $\mu \in \mathcal{L}_{CPL}$ に基づいて修正した新たな知識ベースを $\psi \circ \mu$ と表す. 任意の論理式 $\psi, \mu, \phi \in \mathcal{L}_{CPL}$ に対して, 修正演算子 \circ は以下の公準 (R1)~(R6) によって特徴付けられる:

$$(R1) \quad \psi \circ \mu \models \mu.$$

$$(R2) \quad \psi \wedge \mu \text{ が充足可能ならば, } \psi \circ \mu \equiv \psi \wedge \mu.$$

$$(R3) \quad \mu \text{ が充足可能ならば, } \psi \circ \mu \text{ も充足可能.}$$

$$(R4) \quad \psi_1 \equiv \psi_2 \text{ かつ } \mu \equiv \phi \text{ ならば, } \psi_1 \circ \mu \equiv \psi_2 \circ \phi.$$

$$(R5) \quad (\psi \circ \mu) \wedge \phi \models \psi \circ (\mu \wedge \phi).$$

$$(R6) \quad (\psi \circ \mu) \wedge \phi \text{ が充足可能ならば, } \psi \circ (\mu \wedge \phi) \models (\psi \circ \mu) \wedge \phi.$$

上述の修正の定式化と同様に, Katsuno and Mendelzon による更新の定式化 [Katsuno 92] においても, 知識ベースを \mathcal{L}_{CPL} の 1 個の論理式で表現する. 知識ベースを表す論理式 ψ を, 新情報を表す論理式 $\mu \in \mathcal{L}$ に基づいて更新した新たな知識ベースを $\psi \diamond \mu$ と表す. 任意の論理式 $\psi, \mu, \phi \in \mathcal{L}_{CPL}$ に対して, 更新演算子 \diamond は以下の公準 (U1)~(U8) によって特徴付けられる:

$$(U1) \quad \psi \diamond \mu \models \mu.$$

$$(U2) \quad \psi \models \mu \text{ ならば, } \psi \diamond \mu \equiv \psi.$$

$$(U3) \quad \psi \text{ と } \mu \text{ が共に充足可能ならば, } \psi \diamond \mu \text{ も充足可能.}$$

$$(U4) \quad \psi_1 \equiv \psi_2 \text{ かつ } \mu \equiv \phi \text{ ならば, } \psi_1 \diamond \mu \equiv \psi_2 \diamond \phi.$$

$$(U5) \quad (\psi \diamond \mu) \wedge \phi \models \psi \diamond (\mu \wedge \phi).$$

$$(U6) \quad \psi \diamond \mu \models \phi \text{ かつ } \psi \diamond \phi \models \mu \text{ ならば, } \psi \diamond \mu \equiv \psi \diamond \phi.$$

$$(U7) \quad \psi \text{ が完全ならば, } (\psi \diamond \mu) \wedge (\psi \diamond \phi) \models \psi \diamond (\mu \vee \phi).$$

$$(U8) \quad (\psi_1 \vee \psi_2) \diamond \mu \equiv (\psi_1 \diamond \mu) \vee (\psi_2 \diamond \mu).$$

Katsuno and Mendelzon は更新演算子の意味論的特徴付けを与えている. Katsuno and Mendelzon によると, 更新の操作は意味論的に, 知識ベース ψ のモデルとなるすべての付値関数 $I : \mathcal{L}_{CPL} \rightarrow \{true, false\}$ について, 新情報 μ のモデルで I に最も「近い」付値関数 J をそれぞれ選択し, 更新後の知識ベースのモデルを構成する操作として特徴付けられる.

例 1 テーブルの上に本または雑誌のどちらか一方のみがあるとする. 命題「テーブルの上に本がある」を b , 「テーブルの上に雑誌がある」を m と表すと, この状況を表す知識ベース ψ は論理式 $(b \wedge \neg m) \vee (\neg b \wedge m)$ と同値となる. ここで, 「テ

連絡先: 工藤康生, 室蘭工業大学工学部情報工学科, 〒050-8585
北海道室蘭市水元町 27-1, Tel:0143-46-5469, Fax:0143-46-5499, email:kudo@csse.muroran-it.ac.jp

ブルの上に本がある」との最新情報を得た。この情報 b に基づいて知識ベースを変更する場合、修正を行うか更新を行うかで、得られる知識ベースが異なる:

修正: $\psi \wedge b$ は充足可能なので、公準 (R2) より $\psi \circ b$ は $\psi \wedge b$ すなわち $b \wedge \neg m$ と同値となる。よって、「テーブルの上に雑誌はない」と推論できる。

更新: 公準 (U8) より、 $b \wedge \neg m$ の場合と $\neg b \wedge m$ の場合に分けて考える。その結果、 $\psi \diamond b$ は b と同値になるため、雑誌については何も推論できない。

2.2 拡張論理プログラム

拡張論理プログラム [Gelfond 91] は、失敗による否定 *not* (negation as failure) 及び論理的否定 \neg の 2 種類の否定表現を持つ論理プログラムであり、不完全情報の表現が可能である。

定義 1 拡張論理プログラムは以下の形式のルールからなる集合である:

$$L \leftarrow L_1, \dots, L_m, \text{not } L_{m+1}, \dots, \text{not } L_n \quad (n \geq m \geq 0), \quad (1)$$

ここで、 L 及び L_i ($0 \leq i \leq n$) はリテラルである。 $n = m = 0$ であるルール $L \leftarrow$ をリテラル L と同一視し、ファクトと呼ぶ。

以下、変数を含まないリテラル及びルールを、それぞれ基礎リテラル及び基礎ルールと呼ぶ。また、変数を含むルールは、変数に任意の項を代入して得られるすべての基礎ルールの集合を表すものとする。 Lit を拡張論理プログラムの言語 \mathcal{L} におけるすべての基礎リテラルの集合とする。言語 \mathcal{L} のある原子文 p について、基礎リテラルの部分集合 $S \subseteq Lit$ が $p \in S$ かつ $\neg p \in S$ であるとき、 S は矛盾するという。 S が矛盾しないとき、 S は無矛盾であるという。

拡張論理プログラムの意味論は解集合意味論 (answer set semantics)[Gelfond 91] で与えられる。変数を含まない拡張論理プログラムの解集合は、以下の 2 段階を経て計算される。

定義 2 P は否定による失敗 *not* を含まない基礎ルールの集合とする。 P の解集合 $\alpha(P)$ は、以下の条件を満たす Lit の最小の部分集合である:

1. P に含まれる任意のルール $L \leftarrow L_1, \dots, L_m$ に対して、 $\{L_1, \dots, L_m\} \subseteq \alpha(P)$ ならば $L \in \alpha(P)$ 。
2. $\alpha(P)$ が矛盾するならば、 $\alpha(P) = Lit$ 。

定義 3 P は基礎ルールの集合とする。基礎リテラルの任意の部分集合 $S \subseteq Lit$ に基づいて、 P から *not* を含まない基礎ルールの集合 P^S を以下の手順で構成する:

1. すべての $L_i \in S$ に対して、*not* L_i を含むすべてのルールを P から取り除く。
2. 残りのすべてのルールから項 *not* L_j を削除する。

このとき、 S が P^S の解集合である、すなわち $S = \alpha(P^S)$ であるならば、 S を P の解集合と呼ぶ。

拡張論理プログラム P が無矛盾な解集合を持つとき、 P は無矛盾であるという。式 (1) の形式の基礎ルール R に対して、無矛盾な解集合 S が存在し、 $\{L_1, \dots, L_m\} \subseteq S$ かつ $\{L_{m+1}, \dots, L_n\} \cap S = \emptyset$ ならば $L \in S$ であるとき、基礎ルール R は解集合 S で充足可能であるといい、 $S \models R$ と表す。拡張論理プログラム P のすべての解集合が基礎ルール R を充足するとき、 $P \models R$ と表す。

3. 拡張論理プログラムによる知識ベース

知識ベースを表す古典命題論理の論理式 ψ から、 ψ の意味論的性質を反映した拡張論理プログラム P_ψ を構成する。これ以降、拡張論理プログラムの言語として有限個の原子文から構成される古典命題論理の言語 \mathcal{L}_{CPL} を用いる。

定義 4 以下の手順で、知識ベース ψ から *not* を含まない拡張論理プログラム P_ψ を構成する。基礎リテラルの集合 $S = \{L_1, \dots, L_n\}$ に対して、 $\forall S$ は $\bigvee_{i=1}^n L_i$ を意味するものとする。 $S = \{L\}$ の場合は、 $\forall S$ は L と同一視する:

1. ψ を以下の性質を満たす論理積標準形に変換する:

$$\psi \equiv \bigwedge_{i=1}^m \bigvee D_i,$$

ここで、基礎リテラルの集合 $D_i = \{L_{i1}, \dots, L_{in_i}\}$ は以下の条件を満たす:

1. ψ に現れない原子文のリテラルは D_i に含まれない。
2. $i \neq j$ ならば $D_i \neq D_j$ 。

2. 論理和標準形 $\bigvee D_i = \bigvee_{k=1}^{n_i} L_{ik}$ ($1 \leq i \leq m$) に基づいて、基礎ルールの集合 P_{D_i} を以下の式で構成する:

$$P_{D_i} = \{L_{ij} \leftarrow \bigwedge_{k \neq j} \neg L_{ik} \mid 1 \leq j, k \leq n_i\},$$

ここで、 $\neg L_{ik}$ はリテラル L_{ik} の相補対リテラルである。

3. $P_\psi = \bigcup_{i=1}^m P_{D_i}$ 。

ψ が充足可能であれば、定義 4 で構成された拡張論理プログラム P_ψ に対して、無矛盾な解集合がただ 1 つ存在することは容易に確かめられる。以下、 P_ψ の唯一の無矛盾な解集合を S_ψ と表す。

例 2 例 1 で用いた知識ベース $\psi (\equiv (b \wedge \neg m) \vee (\neg b \wedge m))$ を拡張論理プログラム P_ψ に変換する。まず、 ψ を変換し論理積標準形 $(b \vee m) \wedge (\neg b \vee \neg m)$ を得る。次に、基礎リテラルの集合 $\{b, m\}$ 及び $\{\neg b, \neg m\}$ について、それぞれ基礎ルールの集合を構成する:

$$P_{\{b, m\}} = \{b \leftarrow \neg m, m \leftarrow \neg b\},$$

$$P_{\{\neg b, \neg m\}} = \{\neg b \leftarrow m, \neg m \leftarrow b\}.$$

よって、以下の拡張論理プログラム P_ψ を得る:

$$\begin{aligned} P_\psi &= P_{\{b, m\}} \cup P_{\{\neg b, \neg m\}} \\ &= \{b \leftarrow \neg m, m \leftarrow \neg b, \neg b \leftarrow m, \neg m \leftarrow b\}. \end{aligned}$$

P_ψ は無矛盾であり、唯一の解集合は $S_\psi = \emptyset$ である。

4. 拡張論理プログラムの状態遷移図

本節では、[Kudo 04] で提案された Horn 論理プログラムの状態遷移図を、*not* を含まない拡張論理プログラムに応用する。

定義 5 S をすべての無矛盾なリテラルの部分集合 $S \subseteq Lit$ の集合族とする。*not* を含まない無矛盾な拡張論理プログラム P の各基礎ルール R に対して、 R に関する状態遷移図 ST_R は以下の条件を満たす S 上の二項関係である:

$$(S, S') \in ST_R \Leftrightarrow S \text{ は } P \text{ の解集合, かつ } S' \models R. \quad (2)$$

定義 6 *not* を含まない無矛盾な拡張論理プログラム P に対して, P の状態遷移図 ST_P は以下の式で定義される S 上の二項関係である:

$$ST_P = \bigcap_{R \in P} ST_R. \quad (3)$$

拡張論理プログラム P に含まれるルール R の状態遷移図 ST_R は, プログラム P がルール R を含む限り, R 以外のルール及びファクトをどのように変更しても, 解集合は $(S_\psi, S) \in ST_R$ となる基礎リテラルの集合 S に限定されることを表している. そのため, P の状態遷移図 ST_{P_ψ} は, 5. で提案する修正と更新の使い分けにおいて, P が持つ構文論的な制約を, 「新情報をすべて充足し, かつ P の解集合にできるだけ近い」解集合を選択する意味論的な極小変化に反映させる役割を持つ.

以下の命題 1 は, 知識ベース ψ のモデルと, 拡張論理プログラム P_ψ の状態遷移図 ST_{P_ψ} に現れる, 基礎リテラルの極大無矛盾な部分集合との対応関係を表す.

命題 1 付値関数 $I: \mathcal{L}_{CPL} \rightarrow \{true, false\}$ が知識ベース ψ のモデルであることの必要十分条件は, 基礎リテラルのある部分集合 $S_I \subseteq Lit$ が存在し, 以下の性質を満たすことである:

1. S_I は極大無矛盾である, すなわち, S_I は無矛盾であり, かつ S_I に含まれない任意の基礎リテラル L に対して, $S_I \cup \{L\}$ は矛盾する.
2. $S_I = \{L \in Lit \mid I(L) = true\}$.
3. ψ から構成された拡張論理プログラム P_ψ の状態遷移図 ST_{P_ψ} において, $(S_\psi, S_I) \in ST_{P_\psi}$.

更に, 状態遷移図 ST_{P_ψ} に現れる基礎リテラルの極大かつ無矛盾な部分集合と, 拡張論理プログラム P_ψ の唯一の解集合 S_ψ との間に以下の関係が成り立つ.

命題 2

$$S_\psi = \bigcap \{S \in \mathcal{S} \mid S \text{ は極大無矛盾かつ } (S_\psi, S) \in ST_{P_\psi}\}. \quad (4)$$

例 3 例 2 で構成した拡張論理プログラム $P_\psi = \{b \leftarrow \neg m, m \leftarrow \neg b, \neg b \leftarrow m, \neg m \leftarrow b\}$ の状態遷移図 ST_{P_ψ} を構成する. 以下, 言語 \mathcal{L}_{CPL} における原子文は b と m のみであるとす. 基礎リテラルの無矛盾な集合すべてからなる集合族 \mathcal{S} は以下の 9 個の集合を含む:

$$\begin{aligned} S_0 &= \emptyset, & S_1 &= \{b\}, & S_2 &= \{m\}, \\ S_3 &= \{\neg b\}, & S_4 &= \{\neg m\}, & S_5 &= \{b, m\}, \\ S_6 &= \{b, \neg m\}, & S_7 &= \{\neg b, m\}, & S_8 &= \{\neg b, \neg m\}. \end{aligned}$$

P_ψ の唯一の解集合は $S_\psi = S_0 = \emptyset$ である. P_ψ に含まれる各基礎ルールに対する状態遷移図は, それぞれ以下のように構成される:

$$\begin{aligned} ST_{b \leftarrow \neg m} &= \{S_0\} \times \{S_0, S_1, S_2, S_3, S_5, S_6, S_7\}, \\ ST_{m \leftarrow \neg b} &= \{S_0\} \times \{S_0, S_1, S_2, S_4, S_5, S_6, S_7\}, \\ ST_{\neg b \leftarrow m} &= \{S_0\} \times \{S_0, S_1, S_3, S_4, S_6, S_7, S_8\}, \\ ST_{\neg m \leftarrow b} &= \{S_0\} \times \{S_0, S_2, S_3, S_4, S_6, S_7, S_8\}. \end{aligned}$$

よって, 以下の状態遷移図 ST_{P_ψ} を得る:

$$ST_{P_\psi} = \{(S_0, S_0), (S_0, S_6), (S_0, S_7)\}.$$

ここで, $S_6 = \{b, \neg m\}$ 及び $S_7 = \{\neg b, m\}$ は \mathcal{L}_{CPL} の極大無矛盾な基礎リテラルの集合であり, それぞれ知識ベース ψ のモデルとなる付値関数 $I: I(b) = true, I(m) = false$ 及び付値関数 $J: J(b) = false, J(m) = true$ に対応する. また, $\{b, \neg m\} \cap \{\neg b, m\} = \emptyset$ となり, P_ψ の解集合とも一致する.

5. 拡張論理プログラムに対する信念変更

充足可能な知識ベース ψ から構成された拡張論理プログラム P_ψ に対して, P_ψ の状態遷移図 ST_{P_ψ} を用いて, 新情報に基づいて修正または更新のどちらを行うかを選択し, 使い分けを行うアルゴリズムを提案する. なお, 新情報は基礎リテラルの充足可能な連言に限定する.

アルゴリズム *BeliefChangeForELP*(P_ψ, ST_{P_ψ}, L)

入力: *not* を含まない拡張論理プログラム P_ψ , P_ψ の状態遷移図 ST_{P_ψ} , 新情報を表す基礎リテラルの連言 $L_1 \wedge \dots \wedge L_n$.

```

1   $\|L_1 \wedge \dots \wedge L_n\| \stackrel{\text{def}}{=} \{S \in \mathcal{S} \mid \{L_1, \dots, L_n\} \subseteq S\};$ 
2   $S' \stackrel{\text{def}}{=} \arg_{S \in \|\|L_1 \wedge \dots \wedge L_n\|} \min |Diff(S_\psi, S)|;$ 
3   $S'' \stackrel{\text{def}}{=} \arg_{S \in \|\|L_1 \wedge \dots \wedge L_n\|} \min |\{R \in P_\psi \mid S \not\models R\}|;$ 
4   $ST \stackrel{\text{def}}{=} \bigcap \{ST_R \mid R \in P_\psi, S'' \models R\};$ 
5  if  $(S_\psi, S') \in ST$  then
6       $S^* = S';$ 
7  else
8       $S^*$  として  $S'$  または  $S''$  のどちらかを選択;
9  end if
10 output  $(P_\psi - \{R \in P_\psi \mid S^* \not\models R\}) \cup \{L_1, \dots, L_n\};$ 

```

ここで, 記法 $Diff(S, S')$ は集合 S と S' との対称差を表す. 対称差は以下の式で定義される:

$$Diff(S, S') = (S - S') \cup (S' - S). \quad (5)$$

また, 任意の有限集合 A に対して, 記法 $|A|$ は集合 A の要素の個数を表す.

アルゴリズム *BeliefChangeForELP* において, Step2 で選択される集合 S' は新情報をすべて充足し, かつ解集合 S_ψ との対称差が最も小さい, すなわち解集合に「最も近い」基礎リテラルの集合である. 一方, Step3 で選択される集合 S'' に関しては, 新情報 $L_1 \wedge \dots \wedge L_n$ をすべて受け入れるために, 拡張論理プログラム P_ψ から失われるルールの個数を極力減らすことが考慮されている. そのため, アルゴリズムの Step5 における条件が満たされる場合は, モデルの変化の極小化と失われるルールの個数の極小化が同時に達成される. しかし, Step5 における条件が満たされない場合は, S' 及び S'' に対して一般的に以下の性質が成り立つ:

$$\begin{aligned} |Diff(S_\psi, S')| &< |Diff(S_\psi, S'')| \text{ かつ} \\ |\{R \in P_\psi \mid S' \not\models R\}| &> |\{R \in P_\psi \mid S'' \not\models R\}|. \end{aligned}$$

これは, Step5 の条件が満たされない場合は, どちらか一方の極小化しか実現できないことを意味するため, 以下のジレンマ的状况に直面することとなる:

- 解集合に「最も近い」モデルを選択するために, 失われるルールの個数の増加を受け入れる.
- 失われるルールの個数を極小にするために, 解集合の必要以上の変化を受け入れる.

以下の定理 1 は、アルゴリズム *BeliefChangeForELP* において、 P_ψ から失われるルールを決定するパラメータである S^* として S'' と S' のどちらを選択するかが、修正と更新の使い分けに対応することを示す。

定理 1 アルゴリズム *BeliefChangeForELP* に充足可能な知識ベース ψ から構成された拡張論理プログラム P_ψ 及び P_ψ の状態遷移図 ST_{P_ψ} 、基礎リテラルの充足可能な連言 $L_1 \wedge \dots \wedge L_n$ を入力し、出力された拡張論理プログラム P' から以下の論理式を構成する:

$$\bigwedge \{R \mid R \in P'\}. \quad (6)$$

1. アルゴリズム *BeliefChangeForELP* において、 S^* として S'' を選択したならば、式 (6) は $\psi \circ (L_1 \wedge \dots \wedge L_n)$ と見なされる、すなわち、式 (6) は修正の公準 (R1)~(R6) を満たす。
2. S^* として S' を選択したならば、式 (6) は更新の公準 (U1)~(U6) 及び (U8) を満たし、 $\psi \diamond (L_1 \wedge \dots \wedge L_n)$ と見なされる。

更新の公準 (U7) を満たさないのは、入力される新情報として基礎リテラルの連言のみを扱うためである。

例 4 アルゴリズム *BeliefChangeForELP* を用いて、例 2 で構成した拡張論理プログラム $P_\psi = \{b \leftarrow \neg m, m \leftarrow \neg b, \neg b \leftarrow m, \neg m \leftarrow b\}$ を基礎リテラル b に基づいて書き換える。集合族 $\|b\| = \{S_1, S_5, S_6\}$ に含まれるすべての集合に対して、 P_ψ の解集合 S_0 と「最も近い」基礎リテラルの集合を得るために、 S_0 との対称差を構成する:

$$\begin{aligned} Diff(S_0, S_1) &= \{b\}, & Diff(S_0, S_5) &= \{b, \neg m\} \\ Diff(S_0, S_6) &= \{\neg b, m\}. \end{aligned}$$

よって、 $|Diff(S_0, S_1)| = 1$ 、 $|Diff(S_0, S_5)| = |Diff(S_0, S_6)| = 2$ となり、 $S' = S_1$ を得る。

一方、 $\|b\|$ に含まれる各集合 S に対して、 P_ψ に含まれ、かつ S で充足されないルールの集合を構成する:

$$\begin{aligned} \{R \in P_\psi \mid S_1 \not\models R\} &= \{R \in P_\psi \mid S_5 \not\models R\} = \{\neg m \leftarrow b\}, \\ \{R \in P_\psi \mid S_6 \not\models R\} &= \emptyset. \end{aligned}$$

よって、 S_6 の場合は失われるルールが存在しないため、 $S'' = S_6$ となる。 S'' に基づいて、状態遷移図 ST を構成する:

$$ST = ST_{P_\psi} = \{(S_0, S_0), (S_0, S_6), (S_0, S_7)\}.$$

$(S_0, S_1) \notin ST$ であるので、 S^* としてどちらかを選択する必要がある。 S_1 すなわち S' を選択した場合は、 b に基づく更新を行うこととなる。その結果、出力される拡張論理プログラム P' を以下のように構成し、更新の結果に対応する新たな知識ベース $\psi \diamond b$ を得る:

$$\begin{aligned} P' &= (P_\psi - \{\neg m \leftarrow b\}) \cup \{b\} \\ &= \{b \leftarrow \neg m, m \leftarrow \neg b, \neg b \leftarrow m, b\}. \\ \psi \diamond b &\equiv \bigwedge \{b \leftarrow \neg m, m \leftarrow \neg b, \neg b \leftarrow m, b\} \\ &\equiv (b \wedge m) \vee (b \wedge \neg m). \end{aligned}$$

これは、新情報 b を状況の何らかの変化を反映した情報と見なすため、テーブルの上に本または雑誌のどちらか一方のみがあ

る状況が保たれているとは限らず、新情報 b : 「テーブルの上に本がある」を得ても、雑誌の場所については何も結論できないことを表している。

これに対し、 S_6 すなわち S'' を選択した場合は、 b に基づく修正を行うこととなる。 S_1 の場合と同様の計算によって、修正の結果に対応する新たな知識ベース $\psi \circ b$ を得る:

$$\begin{aligned} \psi \circ b &\equiv \bigwedge \{b \leftarrow \neg m, m \leftarrow \neg b, \neg b \leftarrow m, \neg m \leftarrow b, b\} \\ &\equiv b \wedge \neg m. \end{aligned}$$

新情報 b は元の知識ベース ψ と矛盾しないことから、新情報 b : 「テーブルの上に本がある」を得たことで、 $\neg m$: 「テーブルの上に雑誌はない」と推論できる。

6. おわりに

拡張論理プログラムとして表現される知識ベースに対して新情報を反映させる場合に、プログラムの意味を規定する解集合を 1 個の状態と見なす状態遷移図を用いて修正と更新を使い分けるアルゴリズムを提案した。拡張論理プログラムの変更について、Sakama and Inoue は修正と更新の使い分けが常に可能であるとは限らないと指摘している [Sakama 03]。本稿の提案アルゴリズムでも、 $S'' = S'$ となる、すなわち構文論的な極小変化と意味論的な極小変化が同じ結果をもたらすことがあるため、この場合は修正と更新の区別は困難である。修正と更新の使い分けが必要となるのは、例 4 のように、知識ベースと新情報は矛盾しないものの、知識ベースの一部のモデルが新情報を充足しない場合である。これは一般的に前節で述べたジレンマ的状况となるため、何らかの決定手段が必要となる。

今後の課題として、提案手法のより一般的な拡張論理プログラムへの応用、および継続的な信念変更 (iterated belief change) への応用等が挙げられる。

参考文献

- [Alchourrón 85] Alchourrón, C., Gärdenfors, G. and Makinson, D.: On the Logic of Theory Change: Partial Meet Contraction and Revision Functions, *Journal of Symbolic Logic*, Vol. 50, pp.510–530 (1985).
- [Gelfond 91] Gelfond, M. and Lifschitz, V.: Classical Negation in Logic Programs and Disjunctive Databases, *New Generation Computing*, Vol. 9, pp.365–385 (1991).
- [Katsuno 91] Katsuno, H. and Mendelzon, A. O.: Propositional Knowledge Base Revision and Minimal Change, *Artificial Intelligence*, Vol. 52, pp.263–294 (1991).
- [Katsuno 92] Katsuno, H. and Mendelzon, A. O.: On the Difference between Updating a Knowledge Base and Revising it, *Belief Revision*, Gärdenfors, P. (ed.), pp.183–203, Cambridge Univ. Press (1992).
- [Kudo 04] Kudo, Y. and Murai, T.: Belief Base Revision of Horn Logic Programs Based on State Transition Diagrams, *Proc. of SCI2004*, to appear.
- [Sakama 03] Sakama, C. and Inoue, K.: An Abductive Framework for Computing Knowledge Base Updates, *Theory and Practice of Logic Programming*, Vol. 3, No. 6, pp.671–713 (2003).