3E9-1

信念改竄によるばれない嘘の生成

Perfect Lie Generation by Belief Alteration

奥野　健一　　高橋　和子
Kenichi Okuno　Kazuko Takahashi

関西学院大学大学院理工学研究科
Graduate School of Science and Technology, Kwansei Gakuin University

This study aims at designing a belief alteration system and proposing an algorithm for the generation of perfect lies. The authors consider a database which contains all of the facts a person to be deceived actually believes, and alter it partly so that the lie is made to be perfect. The authors define "the perfect lie" as the fact s/he never infers to be a lie at that stage. Then the authors find the condition for the perfect lie: the world of her/his belief is either coherent or incoherent but s/he does not aware of that fact when s/he believes the lie and does inference from it. The proposed algorithm is sound and terminates.

1. はじめに

論理的にばれない嘘とはどのような性質をもった嘘なのだろうか．嘘に関する研究は様々な分野で行われているが [箱田 06]，どのような嘘がばれない嘘なのかを論理的な観点から追及したような研究は見当たらない．そこで本研究では，ばれない嘘を「ある特定の相手に現時点で論理的にはばれない嘘」とし，その条件を「相手がその嘘を信念に追加し，それに伴う推論を行っても，信念内で矛盾が生じない，あるいは論理的には矛盾が生じていてもその矛盾に気付かないこと」と仮定する．そして，ばれない嘘の条件に形式的な記述を与えた上で，つきたい嘘がばれない嘘の条件を満たすようにするためのアルゴリズムを提案し，その健全性，停止性を証明する．ただし，ここでの矛盾というのはあくまで相手の信念内での矛盾であり，真実との間に生じる矛盾とは区別される．また，「気付かない」とは，人間は論理的全知 (logical omniscience) ではないので，たとえ論理的には矛盾が生じていたとしても当人はそのことに気付かない場合もある，ということを示している．

ばれない嘘について不確定要素を仮定することなしに議論するために，本研究では，嘘をつく相手が現実において信じている事，知っている事の全てが論理式の形で記述されたデータベースを仮定する．これにより，相手に何か（嘘も含む）を信じさせることはデータベースの改竄という処理に置き換えることができるので，「つきたい嘘をばれない嘘にする」ということを「つきたい嘘がばれない嘘の条件を満たすようにするためにデータベースを改竄する（信念改竄）」という処理として捉えることができる．

信念改竄は信念修正 (Belief Revision)[Alchourrón 85] の変種と考えることができる．本研究の特徴は，演繹だけでなくアブダクションに対する考慮も行っている点，人間に何かを信じさせることの難しさを緩和させる試みを行っている点，人間の行う推論の深さに明確な基準を設けている点，などである．これらには信念を追加することによって目的を達成するような処理も含まれており，この考え方は Additive Consolidation [Olsson 98] と類似のものと言える．なお，信念修正の目的は信念を正しい方向に向かわせるための整合化であり，本研究のように間違いだとわかっていることを正当化するためのものではない．このような理由で，本研究では「改竄」という言葉を用いている．

以下，2 節では本研究で仮定する信念世界について，3 節では人間の推論能力について，それぞれ形式的な定義を行う．4 節では，つきたい嘘をばれない嘘にするための信念改竄システムを形式的に記述した上で，その具体的な手続きの一例としてばれない嘘生成アルゴリズムを提案し，5 節でまとめる．

連絡先: 奥野　健一，関西学院大学大学院理工学研究科情報科学専攻高橋研究室，〒 669-1337 兵庫県三田市学園二丁目一番地，TEL:079-565-8300，o.kenichi@ksc.kwansei.ac.jp

2. 信念世界の表現

嘘をつく相手が現実において信じている事，知っている事の全てが論理式として記述されたデータベースをそれぞれ論理式集合 $\mathbf{B}, \mathbf{K}$ とする．なお，$\mathbf{K} \subseteq \mathbf{B}$，$\{\psi \mid (\psi \in \mathbf{B}), (\neg\psi \in \mathbf{B})\} = \emptyset$ である．論理式 (formula) は以下のように定義する．ただし，$\rightarrow$ は関係性を表現する論理結合子で，第 1 引数が第 2 引数の十分条件であることを意味する．今回はこのような単純なシステムを考えるが，連言などの導入による拡張も可能である．

$$
\begin{array}{ll}
formula & ::= fact \mid rule \\
fact & ::= proposition \mid \neg proposition \\
rule & ::= fact \rightarrow fact \mid \neg(fact \rightarrow fact) \\
proposition & ::= \text{命題変数}
\end{array}
$$

3. 人間の推論能力

人間の論理的全知は仮定しないものとする．それゆえ，$\{\psi, (\psi \rightarrow \chi)\} \subseteq \mathbf{B}$ であっても $\chi \in \mathbf{B}$ とは限らない．これでは，信念を追加されたことをきっかけに相手がどこまで深く推論を行うかを推定することができないので，本研究では，人間の推論に対して「推論の過程で既に気付いている事に到達した場合，それ以後の推論は行わない」という仮定を設ける．この仮定の意図は，「既に信じている事を起点とする推論は，過去（それが信念に追加された際）に行っているので，ここで再度行うことはない」ということである．ここで，制限付きの演繹を表現するための次のような表記法を定義しておく．(ψ_i は論理式)

表記	意味
$\mathbf{B}, \psi_0 \rightsquigarrow \psi_m$	$(\psi_0 \rightarrow \psi_1), (\psi_1 \rightarrow \psi_2), \cdots, (\psi_{m-1} \rightarrow \psi_m) \in \mathbf{B}$
$\mathbf{B}, \psi_0 \rightsquigarrow_{\mathsf{T}} \psi_m$	$\psi_0, \psi_1, \cdots, \psi_m \in \mathbf{B}$ $(\psi_0 \rightarrow \psi_1), (\psi_1 \rightarrow \psi_2), \cdots, (\psi_{m-1} \rightarrow \psi_m) \in \mathbf{B}$
$\mathbf{B}, \psi_0 \rightsquigarrow_{\mathsf{NT}} \psi_m$	$\psi_0, \psi_1, \cdots, \psi_m \notin \mathbf{B}$ $(\psi_0 \rightarrow \psi_1), (\psi_1 \rightarrow \psi_2), \cdots, (\psi_{m-1} \rightarrow \psi_m) \in \mathbf{B}$

$\rightsquigarrow_{\mathsf{T}}$ はある人間が過去に行ったであろう推論を表現するために用いる．論理的全知でないことは「たとえ論理的には推論可能な事であっても，その人の現在の信念世界にそれが存在しないのであれば，その人はその事に気付いていない（推論できていない）.」と解釈できる．それゆえ，そのような人間に可能であった推論は $\rightsquigarrow_{\mathsf{T}}$ に対応していると言える．

$\rightsquigarrow_{\mathsf{NT}}$ はある人間の信念に変更を加えた際にその人の信念世界が（その人の推論によって）どのように変化するのかを推定してその結果を表現するために用いる．人間が新たな信念を得た際に行う推論がどの程度の深さまでなのかは推定不可能であるので，この際に限っては論理的全知を仮定せざるを得ないが，本研究では人間の推論に対して上述の仮定を設けているので，これはまさに $\rightsquigarrow_{\mathsf{NT}}$ に対応している．

4. ばれない嘘生成のための信念改竄

4.1 方法

本研究では，嘘をつく相手が現実において信じている事，知っている事の全てが論理式の形で記述されたデータベース $\mathbf{B}, \mathbf{K}$ を仮定するので，この $\mathbf{B}$ を改竄することによってつきたい嘘をばれない嘘にする．

つきたい嘘を論理式 φ とすると，φ が嘘だとばれないためには，$\neg\varphi^{*1}$を信じさせてはならない．そこで本研究では，嘘をつく相手の行う推論として演繹とアブダクションを仮定し，それらによって $\neg\varphi$ が推論されないように対策を行う．演繹対策では $\neg\varphi$ を演繹させた可能性のある仕組みを除去し，アブダクション対策ではもともと φ を唯一の説明としていたようなものに対して他の説明を与える．また，説得力の観点から，つきたい嘘の否定が正しいと仮定した場合に矛盾が導かれる（背理法による証明）ようにするための改竄も行う．これは，人間は自分の信じていることをなかなか曲げようとはしないもので，それゆえ，相手の信念改竄を現実で実践するにはそれなりの説得力が必要である，ということを考慮したものである．さらに，相手に何かを信じさせた（改竄の現実世界での実現）際に，それを起点として相手は推論を行うと考えられるので，その推論（ここでは必然的推論である演繹のみを仮定）を推定し，その結果を $\mathbf{B}$ に反映させる処理も盛り込む．これにより，信念改竄の結果は，信念改竄に伴う推論を相手が行った後でも矛盾が生じていない，あるいは論理的には矛盾が生じていてもそのことに相手は気付いていない，ということを満たす．

4.2 ばれない嘘の条件

つきたい嘘（φ）をばれない嘘として相手の信念世界に追加するための信念改竄システムは，以下のように定義できる．

$$\langle\, \mathbf{L}, \mathbf{B}, \mathbf{K}, \mathbf{C}, \rightsquigarrow, \rightsquigarrow_{\mathsf{T}}, \rightsquigarrow_{\mathsf{NT}}, \dot{+} \,\rangle$$

$\mathbf{B}, \mathbf{K}, \rightsquigarrow, \rightsquigarrow_{\mathsf{T}}, \rightsquigarrow_{\mathsf{NT}}$ については上述の通りとする．$\mathbf{L}$ は全ての論理式からなる集合で，$\mathbf{K} \subseteq \mathbf{B} \subseteq \mathbf{L}$ である．$\mathbf{C}$ は $\mathbf{B}$ に対する追加候補の集合で，$\mathbf{C} \subseteq \mathbf{L}$ である．信念改竄オペレータ $\dot{+}$ は $2^{\mathbf{L}} \times \mathbf{L}$ から $2^{\mathbf{L}}$ への写像で，$\mathbf{B}', \mathbf{B}^-, \mathbf{B}^+$ をそれぞれ $\mathbf{B} \dot{+} \varphi$, $\mathbf{B} \setminus (\mathbf{B} \cap \mathbf{B}')$, $\mathbf{B}' \setminus (\mathbf{B} \cap \mathbf{B}')$ の略記とすると，以下の条件を満たすものである．

($\dot{+}$1) $\varphi \in \mathbf{B}'$.

($\dot{+}$2) $\mathbf{K} \subseteq \mathbf{B}'$.

($\dot{+}$3) $\mathbf{B}^+ \subseteq \mathbf{C}$.

($\dot{+}$4) $\{\psi \mid (\psi \in \mathbf{B}'),\ (\neg\psi \in \mathbf{B}')\} = \emptyset$.

($\dot{+}$5) $\{(\psi, \chi) \mid (\psi \in \mathbf{B}^+),\ ((\mathbf{B}' \setminus \{\psi\}), \psi \rightsquigarrow_{\mathsf{NT}} \chi)\} = \emptyset$.

($\dot{+}$6) $\{(\psi, \chi) \mid (\chi \in \mathbf{B}^-),\ (\psi \neq \chi),\ (((\mathbf{B}' \setminus \{\neg\chi\}) \cup \{\chi\}), \psi \rightsquigarrow_{\mathsf{T}} \chi)\} = \emptyset$.

($\dot{+}$7) $\{\psi \mid (\mathbf{B}', \neg\varphi \rightsquigarrow_{\mathsf{NT}} \psi),\ (\neg\psi \in \mathbf{B}),\ (\neg\psi \in \mathbf{B}')\} \neq \emptyset$.

($\dot{+}$8) $\{\chi \mid (\mathbf{B}, \neg\varphi \rightsquigarrow_{\mathsf{T}} \chi),\ (\{\psi \mid (\mathbf{B}', \psi \rightsquigarrow_{\mathsf{T}} \chi),\ (\psi \neq \chi \neq \neg\varphi)\} = \emptyset),\ (\mathbf{B}', \neg\varphi \rightsquigarrow \chi),\ (\chi \in \mathbf{B}')\} = \emptyset$.

($\dot{+}$2) は「知識であるような信念は改竄してはならない.」ということを要求している．($\dot{+}$5) は「何かを追加したなら，それを起点とした演繹を行わなければならない．ただし，もともと信じているものに到達した場合は，それ以降の演繹は行わない.」を意味し，これは信念追加に伴う相手の推論を推定することに相当する．($\dot{+}$6) は「除去するには，それを演繹させた可能性のある仕組みも壊さなくてはならない.」で，演繹対策を任意の信念除去に伴わせることを要求している．($\dot{+}$7) は「つきたい嘘の否定を背理法によって証明できるようにしなくてはならない.」を，($\dot{+}$8) は「つきたい嘘の否定を唯一の説明とするものに対して，他の説明を用意するか，それが不可能ならそれ自体を除去するか，を行わなくてはならない.」（アブダクション対策）を，それぞれ要求している．そして，上記の 8 個の条件こそが,「$\mathbf{B}$ を改竄した結果として φ がばれない嘘になるための（十分）条件」の形式的記述である．次小節で提案するばれない嘘生成アルゴリズムの実行 $\mathit{GeneratePerfectLie}(B, K, C, \varphi)$ は，$\mathbf{B} \dot{+} \varphi$ の具体的な手続きの一例に相当する．

4.3 ばれない嘘生成アルゴリズム

ばれない嘘生成アルゴリズムは，つきたい嘘がばれない嘘の条件を満たすようにするための信念改竄アルゴリズムである．アルゴリズムを構成する Function の呼び出し関係は図 1 のようになっており，Function1*GeneratePerfectLie* をメインとして各 Function を呼び出す形となっている．Function2*Add*, Function4*Delete* はそれぞれ信念の追加，除去を条件を満たすようにするための手順を踏まえた上で行う手続きで，本アルゴリズムで実行する追加，除去処理は全てこれらを呼び出すことによって行われている．Function3*InferDeduction* は，信念の追加に伴う相手の推論を推定して $\mathbf{B}$ に反映するためのもので，*Add* から呼び出される．Function5*PreventDeduction* は，演繹対策に相当するもので，*Delete* から呼び出される．Function6*ForRefutation* は背理法によって φ を証明できるようにするためのもので，Function7*PreventAbduction* はアブダクション対策に相当するものである．なお，各アルゴリズムに登場する TL はタブーリストを意味し，このリストに追加されたものはその後の処理において $\mathbf{B}$ への追加も除去もされないことが保証されるものとしている．

このアルゴリズムの実行 $\mathit{GeneratePerfectLie}(B, K, C, \varphi)$ が前小節で示した全ての条件を満たすこと（健全性）と，$\mathbf{B}$ を有限集合と仮定した場合の停止性は証明済みである．

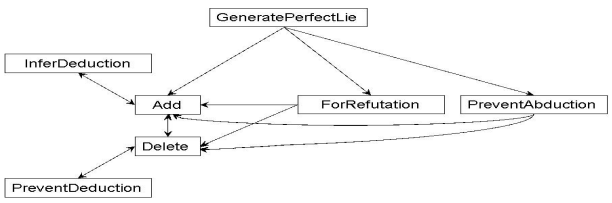

図 1: Function の呼び出し関係

*1 φ の接頭辞が $\neg$ の場合はその $\neg$ を取り除いたものを意味する．

- アルゴリズムの表記上の注意
  - **Formulas** の受け渡しはアドレス渡しに相当する．
  - *Copyof* は，その引数である集合の内容全てをコピーした集合を生成して，そのアドレスを返す Function である．
  - if 節の条件文中に Function の呼び出しが含まれている場合，実際にそれを実行し，その戻り値を条件文の真偽判定に使用する．
  - **return false** は，**その Function の実行でなされた変更を全て取り消して**，呼び出し元に **false** を返して終了する，ということを意味する．(バックトラッキングに相当)
  - **return null** は失敗終了を意味する．

Function 1 *GeneratePerfectLie*

```
INPUT  : Formulas B_0, K, C, Formula φ
OUTPUT : Formulas
B := Copyof(B_0)
TL := {}
if Add(B, K, C, TL, φ) then
    if ForRefutation(B_0, B, K, C, TL, ¬φ) then
        if PreventAbduction(B, K, C, TL, ¬φ) then
            return B
        end if
    end if
end if
return null
```

Function 2 *Add*

```
INPUT  : Formulas B, K, C, TL, Formula ψ
OUTPUT : Boolean
if ψ ∉ C then
    return false
end if
if ψ ∈ TL then
    return false
end if
TL := TL ∪ {ψ}
if ψ ∉ B then
    B := B ∪ {ψ}
    if not InferDeduction(B, K, C, TL, ψ) then
        return false
    end if
    if ¬ψ ∈ B then
        if not Delete(B, K, C, TL, ¬ψ) then
            return false
        end if
    end if
end if
return true
```

Function 3 *InferDeduction*

```
INPUT  : Formulas B, K, C, TL, Formula ψ
OUTPUT : Boolean
while ((ψ → χ) ∈ B) and (ψ ≠ χ) であるような全ての χ に対して do
    if χ ∉ B then
        if not Add(B, K, C, TL, χ) then
            return false
        end if
    end if
end while
return true
```

Function 4 *Delete*

```
INPUT  : Formulas B, K, C, TL, Formula ψ
OUTPUT : Boolean
if ψ ∈ K then
    return false
end if
if ψ ∈ TL then
    return false
end if
TL := TL ∪ {ψ}
if ψ ∈ B then
    B := B \ {ψ}
    if not PreventDeduction(B, K, C, TL, ψ) then
        return false
    end if
    if not Add(B, K, C, TL, ¬ψ) then
        return false
    end if
end if
return true
```

Function 5 *PreventDeduction*

```
INPUT  : Formulas B, K, C, TL, Formula ψ
OUTPUT : Boolean
while ((χ → ψ) ∈ B) and (χ ∈ B) and (χ ≠ ψ) であるような全ての χ に対して do
    if not Delete(B, K, C, TL, (χ → ψ)) then
        if not Delete(B, K, C, TL, χ) then
            return false
        end if
    end if
end while
return true
```

Function 6 *ForRefutation*

```
INPUT  : Formulas B_0, B, K, C, TL, Formula ψ
OUTPUT : Boolean
while ((ψ → χ) ∈ B) or ((ψ → χ) ∈ C) であるような全ての χ に対して do
    if Add(B, K, C, TL, (ψ → χ)) then
        if (¬χ ∈ B_0) and (¬χ ∈ B) then
            return true
        else
            if Delete(B, K, C, TL, χ) then
                if ForRefutation(B, K, C, TL, χ) then
                    return true
                end if
            end if
        end if
    end if
end while
return false
```

Function 7 *PreventAbduction*

```
INPUT  : Formulas B, K, C, TL, Formula ψ
OUTPUT : Boolean
while ((ψ → ω) ∈ B) and (ω ∈ B) and (ω ≠ ψ) であるような全ての ω に対して do
    while (((χ → ω) ∈ B) or ((χ → ω) ∈ C)) and (χ ≠ ω ≠ ψ) であるような全ての χ に対して do
        if Add(B, K, C, TL, (χ → ω)) and Add(B, K, C, TL, χ) then
            次の ω へ.
        end if
    end while
    if not Delete(B, K, C, TL, ω) then
        if not Delete(B, K, C, TL, (ψ → ω)) then
            return false
        end if
    end if
end while
return true
```

5. おわりに

嘘をつく相手の信念が全て記述されたデータベースを仮定し，そのデータベースにつきたい嘘を追加しても矛盾が生じない，あるいは論理的には矛盾が生じていてもそのことに相手が気付かないようにするための信念改竄システムと，その具体的なアルゴリズムを提案した．

今後は，他の論理結合子の導入や，アルゴリズムの効率化など，未成熟なシステムの改良を続けていく予定である．

参考文献

[Alchourrón 85] Carlos E. Alchourrón, Peter Gärdenfors, David Makinson: On the Logic of Theory Change: Partial Meet Contraction and Revision Functions, *The Journal of Symbolic Logic*, Vol. 50, No. 2, pp. 510-530 (1985).

[Fagin 95] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, Moshe Y. Vardi: *Reasoning About Knowledge*, The MIT Press (1995).

[箱田 06] 箱田裕司, 仁平義明: 嘘とだましの心理学 戦略的なだましからあたたかい嘘まで, 有斐閣(2006).

[井上 92] 井上克巳: アブダクションの原理, 人工知能学会論文誌, Vol. 7, No. 1, pp. 48-59 (1992).

[Olsson 98] E.J. Olsson: Making Beliefs Coherent, *Journal of Logic, Language, and Information*, Vol. 7, pp. 143-163 (1998).