類似性の観察に基づく 知識ベースの内包的エラー修正法

Similarity-Driven Knowledge Revision for Intensional Errors

大久保 好章

Yoshiaki OKUBO

北海道大学 大学院工学研究科 電子情報工学専攻

Division of Electronics and Information Engineering, Hokkaido University

yoshiaki@db-ei.eng.hokudai.ac.jp

森田 展博

Nobuhiro MORITA

(同 上)*1

morita@db-ei.eng.hokudai.ac.jp

原口 誠

Makoto HARAGUCHI

(同 上)

makoto@db-ei.eng.hokudai.ac.jp

keywords: knowledge revision, machine learning, similarity, intensional error, goal-dependent abstraction

Summary

In this paper, we propose a new framework of knowledge revision, *Similarity-Driven Knowledge Revision*. For an object-oriented knowledge base $\mathcal{KB}$, our revision is triggered when a *similarity* between sort concepts detected from $\mathcal{KB}$ does not fit a user's intuition. We revise $\mathcal{KB}$ into a knowledge base from which such an undesirable similarity is not detected and in which the logical semantics of $\mathcal{KB}$ is still preserved. An observation of undesirable similarity is due to an *over-general typing of variable* in $\mathcal{KB}$. In order to modify the typing, we introduce a notion of *extended sorts* that can be viewed as a sort concept not appearing explicitly in the original knowledge base. If a variable typing with some sort is considered over-general, the typing is modified by replacing it with more specific extended sort. Such an extended sort can be efficiently identified by forward reasoning with *SOL-resolution* from the original knowledge base. Some experimental results show that the use of SOL-resolution can drastically improve the computational efficiency.

1. はじめに

本稿では、知識の修正・更新問題を、従来とは異なる内包的な観点から議論することで、これまでにない新たな知識修正の枠組を提案する。

知識修正は、知識表現や機械学習研究における主要なテーマのひとつである。この問題に対し、従来の研究では主に外延的アプローチがとられてきた。すなわち、知識ベースの外延（そこから導出可能な事実集合）とユーザが望む外延との不一致が観測された時、その知識ベースはエラー（外延的エラー）を含むとされ、修正処理が起動される。そして、ユーザが望む外延が正しく反映される様に修正が施される。このような修正は、ユーザが望む外延を正例・負例により定めることで、それらの例と無矛盾な仮説を生成する学習問題として定式化される。例えば、文献 [Wrobel 94] における伝統的な知識修正では、知識ベースと論理的に矛盾するいくつかの例がトリガーとな

り修正処理が起動され、それらと無矛盾な外延を有する知識ベースへと修正が試みられる。その結果、所与の例集合との不整合さからくる知識ベースの矛盾が取り除かれる。

この様に、従来の伝統的なアプローチによる知識修正においては、知識ベースのエラーや修正結果（仮説）の正当性は、すべて外延により規定される。しかし、我々が認識する知識ベースの不具合は、その外延的エラーのみではない。つまり、外延的な誤りは認められないが、何らかの不具合を認識し、修正の必要性を感じることもある。当然のことながら、この場合は従来の外延的アプローチでは対処できない。本稿では、こうした外延的に観測することのできないエラーを内包的エラーと呼び、その修正アルゴリズムを与えることで、従来とは異なる内包的な観点からの知識修正の枠組を提案する。

知識ベースには様々なタイプの内包的エラーが考えられる。例として、論理プログラムで記述される知識ベースを考えてみよう。ユーザが望む外延（モデル）を有する

*1 現在、富士通（株）。

論理プログラムは一般には一意に定まらず、意味的 (外延的) に等価な複数の論理プログラムが存在する^{*1}。例えば外延的な誤りは無くとも、それらの中にはユーザにとって望ましくないものもあり、その様な知識ベースは何らかの意味で内包的エラーを含んでいると言える。例えば、ユーザができるだけ簡潔な知識ベースを望む場合、不用な節の存在は一種の内包的エラーとなる。あるいは、節自体は必要な場合でも、それが冗長なボディアトムを含むものであれば、それらアトムの存在はやはり内包的エラーと考えられる。また、詳細は後に議論するが、本稿で扱う知識ベースは、その特殊な具体例として論理プログラムを含むものであり、そこでは、これらエラーに加え、変数の型付けに関する内包的エラーも考えられる。この様に、内包的エラーと呼べるものは様々であり、すべてを統一的に扱うことは困難である。以下では、本稿で修正対象とする内包的エラーを明確にしていこう。

本稿で扱う知識ベースは、順序ソート論理 [Walter 88] により記述された、概念階層・型宣言・領域理論からなるものとする。領域理論は、ある概念 (ソートと呼ぶ) を用いて変数に型を指定することで、その変数が取り得る値を制約した型付き論理式からなり、各論理式はその型付けに用いられたソートが持つ内包的性質を記述したものと捉えることができる。この様な内包的性質は、知識ベースに反映されるソート間の類似性と密接に関連している。

文献 [Gentner 83, Falkenhainer 89, Goldstone 91] に代表される多くの類推研究が指摘する通り、類似性は、概念が有する性質や関係の共有を重視した内包的な観点から論じられるべきであり、これに従うと、知識ベースに記述された内包的性質の共有を根拠に、ソート間の類似性を考えることができる。ここで、ソートが有する内包的性質にエラーが含まれている状況を考えよう。こうしたエラーもまた、知識ベースの内包的エラーのひとつとなることに注意したい。誤りのある内包的性質の共有を根拠として得られる類似性は、当然ながら望ましいものであるとはいえない。つまり、知識ベースが内包的性質に関するエラーを含む場合、そこからは望ましくない類似性が検出されて然るべきである。このことは、従来の外延的アプローチによるそれとは異なる、新たな知識修正の枠組が考えられることを示唆している。すなわち、知識ベースに反映された不適切な類似性の観察をトリガーとする、知識ベースの内包的エラーを除去するための知識修正である。本稿では、この枠組を類似性駆動型知識修正 (*Similarity-Driven Knowledge Revision*) と呼び、その実現アルゴリズムのひとつを与える。

ソートの内包的性質は、そのソートで型付けされた知識により定まることから、知識ベースの内包的エラーの実体として、ここでは、知識の型付けに関する不具合を

考える。いま、ソート概念『電話』と『タバコ』を考えてみよう。どちらの概念も『電車で乗車中は自粛すべき』という (常識的に考えれば) 論理的に正しい内包的性質を有しており、それらは例えば、

$to_be_refrained(Y : phone)$

$\leftarrow on_train_with(X : person, Y : phone)$,

$to_be_refrained(Y : cigarette)$

$\leftarrow on_train_with(X : person, Y : cigarette)$

なる型付き変数を含むふたつの論理式で表現できる^{*2}。この性質の共有を根拠として、『電話』と『タバコ』の類似性を考えることができるが、人によってはこの類似性に違和感を覚える場合がある。ここで、『電話』という概念の捉え方について考えてみよう。『電話』には従来の回線電話もあれば携帯電話もあり、『電話』に対してどちらを強くイメージする (デフォルトと考える) かは、もちろん人それぞれである。例えば、若い年齢層においては、携帯電話は今や誰もが所有する一般的なものであり、特殊な電話であるという認識は全くないであろう。日常での利用形態・頻度から、彼らにとって『電話』とはすなわち携帯電話のことを意味し、回線電話はむしろ特殊なものと捉えられるに違いない。逆に、『電話』の典型はあくまでも回線電話であると考えた人にとっては、携帯電話は特殊な電話のひとつと認識されるだろう。こうした人にとって、先の車内自粛に関する性質は、その人の考える典型的な電話に当てはまるものではないことから、例えばそれが論理的 (外延的) に正しい知識であっても、『電話』の性質としては認め難く、その共有を根拠とする『電話』と『タバコ』の類似性は違和感を与えるものとなろう。その人にとって、先の性質は『電話』の性質ではなく、あくまでも特殊な電話の性質として認識されるのである。これは、もとの論理式における『電話 (*phone*)』による変数の型付けが一般的過ぎることを意味する。このことが原因による望ましくない類似性の検出を抑制するために、本稿では、一般的過ぎる型付けの特殊化により内包的性質の修正を試みる。

詳しい議論は後の章に譲るが、電話の例を用いて簡単に述べると、もとの知識の外延に変化をもたらさない *phone* の特殊化 (詳細化) ソート $phone_{\alpha}$ を同定し、もとの知識の型付けを $phone_{\alpha}$ による型付けに変更する。新たに導入すべき特殊化ソート $phone_{\alpha}$ は、原子式の連言で与えられる拡張ソートにより定義される。ここで、拡張ソートとは、その連言を満たす個体をインスタンスとする概念であり、もとの知識ベースには陽に現れないものである。こうした修正により、もとの知識の論理的意味 (外延的意味) を保ったまま、型付けだけが特殊化され、その結果、修正後の知識ベースにおいては、*phone* と *cigarette* の類似性は検出されず、代わりに *phone* の特殊化である $phone_{\alpha}$ と *cigarette* の類似性が検出される。

拡張ソートの具体的な計算は生成一検査法による。候

*1 論理プログラムの等価性に関する議論については、例えば文献 [Lifschitz 01] を参照されたい。

*2 知識表現の詳細については次章で述べる。

補となる拡張ソートは、もとの知識ベースからの前向き推論により生成されるが、無駄な候補生成を抑制可能な理論的性質を明らかにし、それに基づく効率的な候補生成を、SOL-導出 [Inoue 92] を用いて実現する。

この様に、本稿では、知識ベースに反映された概念間の類似性に対してユーザが感じる違和感を除去するための知識修正手法を考察するが、その意義について簡単に触れておきたい。ここでの修正対象となる知識は、そもそも論理的には誤りがないのであるから、この種の違和感を無視したところで外延的には何ら問題はない。つまり、知識ベースのもとで演繹的な推論のみを行う場合は、あえて修正する必要性はない。しかし、さらに高度な推論である類推 (例えば [Kakuta 97]) を行う場合、その根拠となる類似性に不備があると、妥当な類推結果を得ることができなくなる。先に見た通り、類似性の認識には概念の捉え方が大きく影響し、それは人それぞれ、かつ、一般には時間的にも変化を伴うものである。すなわち、類似性に対するユーザの感覚は静的なものではなく、絶えず変化していると考えるべきである。妥当な類推を行うためには、こうした変化を知識ベースに反映させることが不可欠であり、その具体的方法論を与えることは大変意義深いと著者らは考えている。

本論文の構成は以下の通りである。次章では、本稿での知識表現に用いる順序ソート論理について説明する。3章では、知識ベースに反映される類似性について議論する。それを踏まえ、4章で知識ベースの内包的エラーの意味を明らかにし、その修正手続きの概略を述べ、5章でその詳細を議論する。6章では、実装システムについて紹介し、それにより得られた実験結果について考察する。最後に、本研究のまとめ、および今後の課題について述べる。

なお、本稿は、第2著者が在学中に行なった研究 [森田 99] に基づく発表論文 [Morita 99] を、文献 [原口 99] での議論に沿って加筆・修正したものである。

2. 順序ソート論理における知識表現および推論

本章では、本研究で用いる知識表現の枠組である順序ソート論理 [Walter 88] について述べる。

知識ベースは、ソート階層、型宣言、および領域理論からなるものとする。

ソート階層 $\mathcal{H}$ は、ソートの半順序集合により定義されるものとし、これを $\mathcal{H} = (S, \preceq)$ と表す。 S 中の各ソート s は、ある概念に相当し、その概念の可能なインスタンス (個体) 集合 A_s により解釈される^{*3}。ソート間の順

序関係 $s \preceq s'$ は、 s が s' の下位概念 (s' が s の上位概念) であることを意味し、この時、 s は s' に包摂されると言う。

型宣言 D は、各個体に対してユニークな型を指定するものであり、 $D = \{c_1 : s_1, \dots, c_n : s_n\}$ なる形式で与えられる。ここで、 c_i は個々の個体を表す定数記号、 s_i は S 中のソートであり、 $c_i : s_i$ は、 c_i (に対応する個体) の主型 (primary type) が s_i であることを示す。 c_i の主型は、 $[c_i]$ により参照されるものとする。

個体の主型を用いて、各ソート $s \in S$ の外延 A_s を $A_s = \{c \mid [c] \preceq s\}$ と定義する。 $c \in A_s$ である時、 c は s に属すると言う。 c の主型が s であることと、 c が s に属することの違いに注意して欲しい。この定義から直ちに、 $s \preceq s'$ である時、それら外延間には $A_s \subseteq A_{s'}$ が成り立つことがわかる。

領域理論 $\mathcal{T}$ は、次の形式で記述される関数記号無し型付ホーン節の集合とする。

$$\forall X_1 : s_1 \dots \forall X_n : s_n \quad A \leftarrow B_1 \wedge \dots \wedge B_k.$$

ここで、 A および B_i は正リテラル、 $X_1, \dots, X_n$ はこの節に出現するすべての変数、 $X_j : s_j$ は変数 X_j がソート $s_j \in S$ で型付けされていることを示す型付き変数である。こうした型付けにより、 X_j の取り得る具体的な値は、 s_j に属する個体 (定数) に制限されるものとする。以下では簡単のため、すべての全称限量子を省略し、型付き変数はその節中の最初の出現位置に明記する。例えば、

$$\forall X : s_1 \quad \forall Y : s_2 \quad r(X) \leftarrow p(X, Y) \wedge q(Y)$$

は

$$r(X : s_1) \leftarrow p(X, Y : s_2) \wedge q(Y)$$

と略記する。なお、領域理論中に出現する各定数の主型は、型宣言に記述されているものとする。

型付き論理式は、その型付けに用いられたソートの内包的性質を与えていると考えられる。例えば、節 $q(X : s) \leftarrow p(X)$ は、ソート s が、“ p が成り立てば、必ず q が成り立つ” なる性質を有する概念であることを述べている。

以下、ソート階層 $\mathcal{H}$ 、型宣言 D 、および領域理論 $\mathcal{T}$ からなる知識ベース $\mathcal{KB}$ を $\mathcal{KB} = \langle \mathcal{H}, D, \mathcal{T} \rangle$ と表す。

順序ソート論理における推論は、代入の扱い以外は一階述語論理におけるそれに従う。ここでの代入 (順序ソート代入) θ は $\theta = \{X_1 : s_1/t_1, \dots, X_n : s_n/t_n\}$ なる形式で与えられ、各 t_i は、型付き変数 $Y_i : u_i$ (ただし、 $u_i \preceq s_i$)、あるいは、定数 c_i (ただし、 $[c_i] \preceq s_i$) のいずれかでなければならない。

順序ソート代入による単一化を、順序ソート単一化と呼ぶ。知識ベース $\mathcal{KB} = \langle \mathcal{H}, D, \mathcal{T} \rangle$ のもとでの実際の推論過程では、順序ソート単一化の際、極大な単一化代入

^{*3} ここでは A_s を非空であると仮定する。この仮定は、順序ソート論理における推論と、対応する一階述語論理のそれとの等価性を保証するために必要となる。なお、 A_s の正確な定義は後に与える。

が用いられる．順序ソート単一化を用いて $\mathcal{KB}$ から論理式 F が証明できることを, $\mathcal{T} \vdash_{\mathcal{H}} F$ と表す*4．

3. 概念間の類似性

本章では, 知識ベースに記述された内包的性質の共有に基づくソート間の類似性について議論する．

いま, 知識ベース (領域理論) に節 $q(X : s_1) \leftarrow p(X)$ および $q(X : s_2) \leftarrow p(X)$ が含まれているとしよう．先に述べた通り, これらは, ソート s_1 と s_2 がそれぞれ有する内包的性質を記述したものであり, どちらのソートも共に“ p が成り立つならば, 必ず q も成り立つ”なる性質を有することを示している．よって, この様な内包的性質の共有を根拠として, s_1 と s_2 の類似性を見出すことができるであろう．本稿では, この意味でのソート間の類似性について考えていく．

こうした類似性の検出アルゴリズムのひとつとして, GDA [Kakuta 97, Okubo 98] が挙げられる．GDA は, 目的や文脈に依存した類似性の検出を目指して提案されたアルゴリズムであり, その理論的基礎は, 述語写像に基づく抽象化 [Tenenber 89] の枠組を文脈依存性を取り込むことにより拡張した, ゴールに依存した抽象化 (*Goal-Dependent Abstraction*) [Haraguchi 00, Kakuta 97, Okubo 98] にある．ここでは, 順序ソート論理におけるゴールに依存した抽象化について紹介し, 本稿で扱う類似性の定義を与える．

まず, 一階述語論理における抽象化理論を順序ソート論理に対して拡張した, ソート写像に基づく抽象化について述べる．

3.1 ソート写像に基づく抽象化

順序ソート論理で記述された知識ベースは, ソート写像に基づいて抽象化することができる．

【定義 3.1】(ソート写像)

ソート階層 $\mathcal{H} = (S, \preceq)$ について, S の分割 (partition) $\{S_1, \dots, S_k\}$ を考える*5．各 S_i に対して, S に出現しないソート記号 α_{S_i} を対応させ, それらの集合を $S' = \{\alpha_{S_1}, \dots, \alpha_{S_k}\}$ とする．この時, 次の通り定義される写像 $\psi : S \rightarrow S'$ をソート写像と呼ぶ．

$s \in S_i$ なるソート s に対して, $\psi(s) = \alpha_{S_i}$. ■

なお, 以下の議論では, ソート写像 ψ と対応する S の分割を同一視する．

ソート写像 ψ は, 知識ベース $\mathcal{KB} = \langle \mathcal{H}, \mathcal{D}, \mathcal{T} \rangle$ 中の任意の表現に対する写像に拡張可能である．すなわち, 表現 E について, $\psi(E)$ は, E 中の各ソート s を $\psi(s)$ に書き換えた結果であるとする．例えば, ソートの包摂関係 $s_1 \preceq s_2$ は, $\psi(s_1) \preceq \psi(s_2)$ に写像される．

*4 正確には $\mathcal{D}$ も明示すべきであるが, 表記が繁雑になるので敢えて省略する．

*5 つまり, $S = \bigcup_{i=1}^k S_i$ かつ $S_i \cap S_j = \emptyset (i \neq j)$ である．

S の分割 $\{S_1, \dots, S_k\}$ を定めるソート写像 ψ について, 同じ S_i に属するソート同士は互いに類似していると考えことにしよう．すると, ψ は, ソート間のある類似関係を与える仮説であると見なすことができる．以下ではこうした考えに従い, $s_1, s_2 \in S_i$ である時, つまり, $\psi(s_1) = \psi(s_2) = \alpha_{S_i}$ である時, s_1 と s_2 は ψ のもとで類似しているといい, α_{S_i} は, S_i に属する任意のソートの違いを捨象した抽象ソートであると考える．

ここで, ソート写像が (仮説的に) 定める類似性と, 知識ベースに記述された内包的性質との関係について触れておこう．抽象ソート α_{S_i} は, S_i に属する任意のソートの違いを捨象したソートであるから, α_{S_i} の有する内包的性質は, S_i に属する任意のソートが共有する内包的性質のみであり, 共有されない性質は, 決して α_{S_i} の性質にはなり得ないと考えべきである．このことは, あるソート s の内包的性質が, その抽象ソートにおいて成立するか否かは, s との類似関係を定めるソート写像に依存して決まることを意味する．例えば, ソート写像

$$\begin{aligned} \psi_1 &= \{S_{12} = \{s_1, s_2\}, S_3 = \{s_3\}\} \quad \text{および} \\ \psi_2 &= \{S_{13} = \{s_1, s_3\}, S_2 = \{s_2\}\} \end{aligned}$$

を考えてみる．いま, 知識ベース中に次の論理式が記述されていたとする．

$$q(X : s_1) \leftarrow p(X). \quad (1)$$

$$q(X : s_2) \leftarrow p(X). \quad (2)$$

$$t(X : s_3) \leftarrow p(X). \quad (3)$$

式 (1) が与える s_1 の内包的性質は, ψ_1 のもとで s_1 と類似した s_2 にも共有されていることが式 (2) からわかる．よって, この性質は s_1 および s_2 の抽象ソート $\alpha_{S_{12}}$ においても成立し, 論理式 $q(X : \alpha_{S_{12}}) \leftarrow p(X)$ は妥当なものであると言える．これに対し, ψ_2 のもとで s_1 と類似した s_3 は, この内包的性質を持たない．つまり, この性質は, s_1 および s_3 の抽象ソートである $\alpha_{S_{13}}$ の性質とはなり得ず, 論理式 $q(X : \alpha_{S_{13}}) \leftarrow p(X)$ は妥当とは言えないことになる．式 (3) が与える性質についても同様に考えると, ψ_1 のもとでは抽象ソート α_{S_3} の性質となるが, ψ_2 のもとでは $\alpha_{S_{13}}$ の性質とはならないことがわかる．

この様に, 抽象ソートの内包的性質は, ソート写像が定めるソート間の類似関係に依存して決まる．こうした様子は, ソート写像に基づく抽象化として定式化される．

【定義 3.2】(ソート写像に基づく抽象化)

$\mathcal{KB} = \langle \mathcal{H}, \mathcal{D}, \mathcal{T} \rangle$ を知識ベース, ψ をソート写像とする．領域理論 $\mathcal{T}$ 中の節 C が次の条件を満足する時, C は ψ のもとで抽象化可能であるといい, $\psi(C)$ を C の抽象化と呼ぶ：

任意の $\tilde{C} \in \psi^{-1}(\psi(C))$ について, $\mathcal{T} \vdash_{\mathcal{H}} \tilde{C}$ ．

ここで、任意の表現 E' について、 $\psi^{-1}(E') = \{E \mid \psi(E) = E'\}$ とする。

同様に、ソート階層 $\mathcal{H}$ 中の包摂関係 $Subs = s_1 \preceq s_2$ が以下の条件を満足する時、 $Subs$ は ψ のもとで抽象化可能であるといい、 $\psi(Subs) = \psi(s_1) \preceq \psi(s_2)$ を $Subs$ の抽象化と呼ぶ：

任意の $\tilde{s}_1 \in \psi^{-1}(\psi(s_1))$ について、ある $\tilde{s}_2 \in \psi^{-1}(\psi(s_2))$ が存在し、 $\tilde{s}_1 \preceq \tilde{s}_2 \in \mathcal{H}$ 。 ■

定義 3-2 から、ソート写像 ψ のもとで抽象化可能な節は、 ψ が定めるすべての類似ソートに共有される内包的性質を与えることがわかる。こうした抽象化の枠組を拡張することで、目的や文脈に依存した類似性を定式化することができる。

いま、所与の知識ベースのもとで、ある論理式（ゴール）を証明することを目的と考え、その証明過程で使われる節と包摂関係が文脈を定めるものとする。すなわち、ここでいう文脈とは、目的に關与した内包的性質を与えるものと考えることができる。よって、このような内包的性質の共有を根拠とした類似性は、その目的・文脈に特有なものであるといえる。これを、その目的・文脈における適切な類似性と呼ぶことにしよう。すると、ある文脈を定める（所与のゴールの証明過程で使われる）節と包摂関係を全て抽象化可能とするソート写像は、この目的・文脈における適切な類似性を定めていることになる。このことは、ゴールに依存した抽象化（Goal-Dependent Abstraction）として定式化される。

【定義 3-3】（ゴールに依存した抽象化（類似性））

$\mathcal{KB} = \langle \mathcal{H}, \mathcal{D}, \mathcal{T} \rangle$ を知識ベース、 ψ をソート写像とする。いま、 $\mathcal{KB}$ のもとで、あるゴール G の証明を考え、その証明過程で使われた節集合、および、包摂関係集合をそれぞれ $\mathcal{T}(G)$ 、 $\mathcal{H}(G)$ とする。 ψ が以下の条件を満足する時、 ψ は ゴール G に適切な抽象化（類似性）を定めるといふ。

- (1) 任意の $C \in \mathcal{T}(G)$ について、 C は ψ のもとで抽象化可能である。
- (2) 任意の $Subs \in \mathcal{H}(G)$ について、 $Subs$ は ψ のもとで抽象化可能である。 ■

GDA は、知識ベースとゴール G を入力とし、 G に適切な抽象化（類似性）を定めるソート写像を出力するアルゴリズムである。その詳細については、文献 [Kakuta 97, Okubo 98] を参照されたい。

本稿では、所与の知識ベースに対して GDA が検出する（あるゴールに適切な）類似性を観測し、望ましくない類似性が検出された場合に知識ベースの修正処理が起動される、類似性駆動型知識修正を提案する。これまでみてきた通り、知識ベースに反映された類似性は、そこに記述された内包的性質と密接に関係していることから、望ましくない類似性の検出は、内包的性質に関するエラーに起因すると考えられる。ここでは、こうしたエラーを

除去することで、修正後の知識ベースからは望ましくない類似性が検出されない様にする。

4. 知識ベースの内包的エラー：知識の型付けに関する不具合

本章では、知識ベースに反映されたソート間の類似性とユーザの感覚とのズレについて考察することで、ここで扱う知識ベースの内包的エラーを形式的に定める。さらに、その修正手法の概略を述べることで、後の議論のアウトラインを与える。

いま、コップと缶に関する次の性質を考えてみよう。

$$\text{不燃物}(X : \text{コップ}) \leftarrow \text{金属性}(X). \quad (4)$$

$$\text{不燃物}(X : \text{缶}) \leftarrow \text{金属性}(X). \quad (5)$$

これらは、コップと缶は共に“金属製ならば不燃物である”という性質を共有していることを示し、それ故に、コップと缶の間に類似性を見出すことができる。常識的にみれば、確かにこれら性質は正しいものであり、式 (4)、(5) とともに論理的（外延的）には何ら問題のない知識であるといえる。よって、内包的性質の共有を根拠として類似性を捉える限りは、コップと缶の類似性は妥当なものと言わざるを得ない。しかし、この様に論理的に正しい知識を根拠にしたとしても、ユーザがその類似性に違和感を覚える状況が考えられる。

そこで、コップの性質を記述した式 (4) の意味するところをもう一度考えてみよう。これは、コップの場合、金属性のものは不燃物であると述べているが、実際には様々な材質で作られたコップがある。そうした中、コップという概念に対して、紙性のコップをデフォルトで連想するユーザがいたとする。紙コップは金属性ではないので、このユーザが通常連想するコップに対して式 (4) が使われることはない。つまり、このユーザにとって、式 (4) はほとんど意味をなさない不燃物と考えられ、コップの性質としては理解し難いものとなろう。それに対し、知識ベースは式 (4) をコップの性質として認めており^{*6}、その共有を根拠にコップと缶の類似性が検出される。しかし、これは、このユーザにとっては不燃物を根拠としたものであることから、知識ベースに反映された類似性とユーザの感覚とのズレが生じてしまう。

この様に、内包的性質の共有を根拠とした類似性を考える場合、共有性質が論理的に正しいものであっても、ユーザにとって望ましくない類似性が検出されることがある。本稿では、これも知識ベースにおける不具合のひとつであると考え、その様な類似性が検出されない様に知識の修正を試みる。望ましくない類似性の検出は、そこに記述された内包的性質に関するエラーの存在を示唆していることから、これを本稿で扱う内包的エラーと定

*6 さらに、それは論理的に誤りのない正しい知識である。

め、具体的な修正対象とする。特に、もとの知識ベースはそもそも論理的には何ら問題がないのであるから、修正後もその論理的意味(外延の意味)は保存される手法を提案する。ここで扱う知識ベースの内包的エラーは、そこに反映された類似性により認識され、その論理的意味の誤り(外延的エラー)としては観測できない。よって、正例・負例との整合性に基づく外延的アプローチをとる従来の知識修正手法では、こうした不具合に対処できないことを強調しておく^{*7}。

以下では、内包的エラーの実体を定め、その修正手法の基本アイデアを述べる。

望ましくない類似性の検出を抑制可能な最も素朴な修正は、先の式(4)の様な不用と考えられる知識を知識ベースから取り除くことである。これにより、類似性の根拠となる知識がなくなるのであるから、修正後はコップと缶の類似性は当然検出されなくなる。しかし、知識の除去による修正は一般に外延的な変化を伴い、もとの知識ベースが持つ論理的意味を失うことになるので、本稿で目指す修正手法としては不十分である。そこで、再度先の例に戻ろう。

紙コップに対しては、式(4)は確かにまったく意味のない不用な知識であるが、実際には様々な材質で作られたコップが存在し、当然その中には金属性のものも含まれ、これらコップに対しては、式(4)の存在は意味のあるものとなる。つまり、式(4)が与える性質は、コップの特定の部分概念に対しては不用であるが、別の部分概念に対しては必要不可欠なものとなっている。しかし、もとの知識記述ではその区別がなされておらず、この意味で式(4)は、論理的には正しいが、記述粒度が粗過ぎると言える。すなわち、コップによる一般的過ぎる知識の型付けが原因で、一部の部分概念に対して知識の冗長性が生じている。さらに、ユーザにとってのコップの典型が、この知識が本来不用な部分概念(紙コップ)であったことから、結果として、知識ベースに反映された類似性とユーザの感覚とにズレが生じるのである。

以上のことから、本稿で扱う内包的エラーの実体は、知識の一般的過ぎる型付けであると定める。よって、ここでの修正とは、もとの知識の論理的意味を変えなく、その型付けに用いられたソートを特殊化し、その特殊化ソートでもとの型付けを置き換える操作、つまり、型付けの特殊化を意味する。

先の例を用いて修正処理の概略を述べると、まず、式(4)の論理的意味に変化をもたらさないコップの特殊化

概念 α を同定し、それを用いて式(4)を

$$\text{不燃物}(X:\alpha) \leftarrow \text{金属性}(X)$$

に置き換えることで領域理論の修正を行なう。ここで、式(4)の論理的意味を保存するための自明な解として、 α を『金属性のコップ』と定義することが考えられる。その α を用いることで、ここでの修正処理の目的は確かに達成されるが、得られる知識は、“『金属性のコップ』である個体 X が金属性であるならば、不燃物である” なるものとなり、条件部に冗長な記述を含むことになる。この様な冗長な修正結果を排除するために、本研究では、『金属性のコップ』が必ず有する性質(述語)を前向き推論により導出し、その性質を用いて α を定義する。例えば、『金属性のコップ』が必ずハンドルを有しているならば、 α を『ハンドル付きコップ』と定義する。その結果、 α はコップの下位概念であり、なおかつ『金属性のコップ』の上位概念となることから、 α を用いて型付けを特殊化しても式(4)の論理的意味は保存される、すなわち、修正前後の外延的等価性が保証される。さらに、この α を、既存ソート階層におけるコップの下位の適切な位置に挿入することで階層の更新を行ない、それに伴い、型宣言も更新する。こうした修正後の知識ベースにおいては、コップと缶の類似性は検出されず、代わりに α と缶の類似性が観測されることになる。

5. 型付けの特殊化による内包的エラーの修正

本章では、これまでに述べた一連の修正処理に関して、より詳細な議論を展開する。

$KB = \langle \mathcal{H}, \mathcal{D}, \mathcal{T} \rangle$ を知識ベース、 G をゴールとする。この時、GDA アルゴリズムにより、ソート s と u の類似性が検出されたが、ユーザがそれに違和感を覚えたと仮定する。特にここでは、その原因となる内包的エラーとして、 G の証明に用いられた節集合 $T(G)$ 中のある節 C のボディに、 s による一般的過ぎる型付き変数があることを想定する。前章で述べた通り、この型付けを修正するためには、 s の特殊化ソート s_α を同定する必要がある。求めるべき s_α の満たす条件を正確に与えるために、まず拡張ソート^{*8}を導入する。

5.1 拡張ソート

拡張ソートは、あるアトムの連言を満たす個体をそのインスタンスとして取る概念であり、これにより、既存のソート階層には陽に現れない概念を表すことができる。

【定義 5.1】(拡張ソート)

es をアトム連言とする。いま、 es 中のある型付き変数 $X:s$ に注目し、それ以外の変数は、すべて存在限定されているとする。この時、 es を、 $X:s$ をルート変数

^{*7} ここでの知識ベースを、対応する一階述語論理のそれに変換することで、内包的エラーを外延的エラーとして扱うことができる。よって、原理的には従来の外延的アプローチによる修正が不可能ではないが、そのためには、観測された類似性の不具合を反映した正例・負例を具体的かつ的確に与える必要がある。しかし、類似性に対する感覚的な認識をもとに、このような具体例を用意することは極めて困難なため、実際上は従来のアプローチで対処することは難しい。

^{*8} 文献 [Haraguchi 00] における “Concept Graph” に相当する。

とする拡張ソート (*extended sort*) と呼ぶ。 es のルート変数は $Root(es)$ で参照される。 ■

以下の議論において、 $Root(es) = X : s$ なる拡張ソート es を考える際、そのルート変数を明確にする場合は、 $es(X : s)$ と、それ以外の変数も明確にする場合は、 $es(X : s, X_1 : s_1, \dots, X_n : s_n)$ と記述する。また、拡張ソートを、その構成アトム集合と同一視することもある。

拡張ソートは以下に示す通り、その連言を満たすルート変数の可能なインスタンス集合として解釈される。

【定義 5.2】(拡張ソートの外延)

$\mathcal{KB} = \langle \mathcal{H}, \mathcal{D}, \mathcal{T} \rangle$ を知識ベース、 $es(X : s)$ を拡張ソートとする。 $\mathcal{KB}$ のもとでの es の外延を $E_{\mathcal{KB}}(es)$ で表し、

$$E_{\mathcal{KB}}(es) = \{c \mid c \in A_s \text{ かつ } \mathcal{T} \vdash_{\mathcal{H}} \exists es(c)\}$$

と定義する。ここで、 $\exists es(c)$ は、 $es(c)$ 中のすべての変数が存在限定されていることを表す。 ■

定義 5.2 から直ちに、拡張ソート $es(X : s)$ について、 $E_{\mathcal{KB}}(es) \subseteq A_s$ であることがわかる。すなわち、 es は s の特殊 (下位) 概念に相当する。例えば、拡張ソート $es(X : person) = owns_computer(X : person, Y : personal_computer) \wedge be_installed_in(Z : unix, Y)$ は、『人 ($person$)』なる概念に対し、特に『UNIX システムがインストールされたパソコンを所有する人』を意味する概念に相当する。

ここで『UNIX システムがインストールされたデスクトップパソコンを購入した人』を意味する拡張ソート $es'(X : person) = bought(X : person, Y : desk_top_pc) \wedge be_installed_in(Z : unix, Y)$ を考えてみよう。いま、領域理論がルール $r = owns_computer(X : person, Y : computer) \leftarrow bought(X, Y)$ を含み、かつ、ソート階層において $desk_top_pc \preceq personal_computer \preceq computer$ なる関係が成り立っていると仮定しよう。 r および $desk_top_pc \preceq computer$ より、 $X : person$ がデスクトップパソコン $Y : desk_top_pc$ を購入したなら、 $owns_computer(X : person, Y : desk_top_pc)$ が成り立つ、すなわち、 X は Y を所有することがわかる。さらに、 $desk_top_pc \preceq personal_computer$ であるから、 es' のインスタンスは es のインスタンスでもあることがわかる。つまり、 es' は es の特殊概念に相当すると言える。

この様に、拡張ソート間にも外延の包含関係に基づく包摂関係を考えることが可能であり、このことは次の定義にまとめられる。

【定義 5.3】(拡張ソート間の包摂関係)

$\mathcal{KB} = \langle \mathcal{H}, \mathcal{D}, \mathcal{T} \rangle$ を知識ベース、 es_1 および es_2 を、 $Root(es_1) = Root(es_2) = X : s$ なる拡張ソートとする。ここで、

$$\mathcal{T} \vdash_{\mathcal{H}} \forall X : s (\exists es_2 \leftarrow \exists es_1)$$

である時、 es_1 は es_2 に ($\mathcal{KB}$ のもとで) 包摂されると言い、 $es_1 \preceq_{\mathcal{KB}} es_2$ と表す。ここで、 $\exists es_i$ は、 es_i 中の

ルート変数 $X : s$ 以外のすべての変数が存在限定されていることを表す。 ■

[定理 5.1]

拡張ソート es_1 および es_2 について、次のふたつは同値である。

- (1) $es_1 \preceq_{\mathcal{KB}} es_2$.
- (2) $E_{\mathcal{KB}}(es_1) \subseteq E_{\mathcal{KB}}(es_2)$.

《証明》

(1) $\Rightarrow$ (2): $es_1 \preceq_{\mathcal{KB}} es_2$ であると仮定する。いま、 $\mathcal{KB}$ の最小モデル $\mathcal{M}_{\mathcal{KB}}$ を考える。仮定より、任意の $c \in A_s$ について、 $\mathcal{M}_{\mathcal{KB}}$ のもとで、 $\exists es_1(c)$ が真ならば $\exists es_2(c)$ も真である。よって、 $\mathcal{M}_{\mathcal{KB}}$ のもとで $\exists es_1(c)$ が真ならば、 $\mathcal{M}_{\mathcal{KB}}$ のもとで $\exists es_2(c)$ も真である。すなわち、 $\mathcal{T} \vdash_{\mathcal{H}} \exists es_1(c)$ ならば $\mathcal{T} \vdash_{\mathcal{H}} \exists es_2(c)$ が成立する。これから $c \in E_{\mathcal{KB}}(es_1)$ であれば $c \in E_{\mathcal{KB}}(es_2)$ であることがわかる。

(1) $\Leftarrow$ (2): 上記議論を逆にたどることで証明可能。 □

ソート階層 $\mathcal{H}$ 中のソート s の外延 A_s は、領域理論に依存しないので、知識ベース $\mathcal{KB}$ のもとでの外延であると考えても差し支えない。以下の議論では、 $\mathcal{H}$ 中のソート・拡張ソートを問わず、 $\mathcal{KB}$ のもとでの外延間に包含関係がある時、それらの間には包摂関係 $\preceq_{\mathcal{KB}}$ があるとする。

5.2 特殊化拡張ソート

修正対象となる一般の過ぎる型付き変数 $X : s$ をボディに含む節 C を $C = H \leftarrow B_1 \wedge \dots \wedge B_n$ とする。ここでこの修正目標は、 C の持つ論理的意味を失わないまま、 s による型付けを特殊化することである。

ここで、 $es = B_1 \wedge \dots \wedge B_n$ を、ルート変数が $X : s$ の拡張ソートであると考え、次のふたつの条件を満たすソート s_α を用いれば、目標とする修正が実現できる。

- es は s_α に包摂される。
- s_α は s に包摂される。

後に証明を与えるが、前者により、 s_α を用いた特殊化操作は C の論理的意味に影響を与えない事が保証される。また、後者により、 s_α を s の特殊化 (下位) 概念とみなすことができる。

より正確には、こうした s_α は、次に定義する特殊化拡張ソートにより定められるとする。

【定義 5.4】(特殊化拡張ソート)

修正対象の型付き変数 $X : s$ をボディに含む節 $C = H \leftarrow B_1 \wedge \dots \wedge B_n$ について、拡張ソート $es(X : s) = B_1 \wedge \dots \wedge B_n$ を考える。拡張ソート $ES(X : s)$ が次の4つの条件を満たす時、これを es に関する s の特殊化拡張ソートと呼ぶ^{*9}。

条件 (1) $es \preceq_{\mathcal{KB}} ES$.

条件 (2) $\mathcal{KB}$ のもとで、 ES をゴールとした時の任意

*9 文脈上明らかな場合は、単に特殊化拡張ソートと呼ぶ。

の計算解代入 θ について, $X: s_1/t \in \theta$ なる項 t が存在する.

条件 (3) $ES \cap es = \phi$.

条件 (4) $ES_0 \subseteq ES$ なる任意の ES_0 について, C のヘッド H と ES_0 間には, ルート変数の選び方によらず $H \not\leq_{KB} ES_0$ が成立する. ■

上述した通り, 条件 (1) により, ES による型付けの特殊化操作は, もとの知識の論理的意味には影響を及ぼさないことが保証される. また, 条件 (2) により, ES は s の真の下位ソートに相当する, つまり, $E_{KB}(ES) \subset A_s$ であることが保証される. 条件 (3) と (4) は, 修正結果にある種の冗長性を生じさせないために必要となる. なお, 条件 (2), (3), (4) の必要性について述べた定理を付録に与える.

次節では, これらの条件を満足する特殊化拡張ソートの具体的な計算手続きを与えることにする.

5.3 特殊化拡張ソートの計算

特殊化拡張ソートの計算は, 生成-検査法によるものとする. 特殊化拡張ソートの満たすべき条件 (1) より, es を包摂する拡張ソートを候補として生成し, 条件 (2), (3), (4) を満足するか否かを検査することで, 求めるべき特殊化拡張ソート ES を同定する.

§1 候補の生成

候補となる拡張ソートは前向き推論により生成することができる. 拡張ソート $es = es(X: s, Y_1: s_1, \dots, Y_n: s_n)$ を包摂する拡張ソート $es'(X: s)$ は, 定義 5.3 より,

$$\begin{aligned} T \vdash_{\mathcal{H}} \forall X: s (\exists es'(X: s) \\ \leftarrow \exists Y_1: s_1 \dots \exists Y_n: s_n es(X: s)) \end{aligned}$$

を満たす. これは, 式変形することで

$$\begin{aligned} T \vdash_{\mathcal{H}} \forall X: s \forall Y_1: s_1 \dots \forall Y_n: s_n \\ (\neg es(X: s) \vee \exists es'(X: s)) \end{aligned}$$

と表すことができる. ここで, $X: s$ と $Y_i: s_i$ に対し, KB 中に出現しない定数 $a: s$ および $b_i: s_i$ を代入した es の基礎例 $es(a, b_1, \dots, b_n)$ を考える. すると, 上式は,

$$T \cup \{es(a, b_1, \dots, b_n)\} \vdash_{\mathcal{H}} \exists es'(a) \quad (6)$$

と変形できる.

以上から, es を包摂する拡張ソート, つまり, 特殊化拡張ソートの候補は, KB 中に出現しない定数による es の基礎化代入 θ を用いて, $T \cup \{es\theta\}$ からの前向き推論により生成可能であることがわかる. こうして得られた候補の中で, 定義 5.4 の条件 (2), (3), (4) を満足する拡張ソート ES を, es に関する s の特殊化拡張ソートとする. s による型付けを特殊化する際に用いる s_α は, こうした ES により定義されるソート概念である.

一般に, 生成-検査法による計算処理においては, 無駄のない候補生成が望まれ, 役に立たない(解になり得ない)

候補の生成は極力回避すべきことは言うまでもない. 上述した通り, 特殊化拡張ソートの候補生成は, $T \cup \{es\theta\}$ からの前向き推論により可能であるが, これにより生成される候補数は一般に膨大なものとなり, その中には多くの不要な候補も含まれることが容易に予想される. よって, 効率化が不可欠であると考えられよう. 次に示す定理は, 不要な候補生成の枝刈りに有用な性質を与えるものである.

[定理 5.2]

修正対象の型付き変数 $X: s$ をボディに含む節 $C = H \leftarrow B_1 \wedge \dots \wedge B_n$ について, 拡張ソート $es(X: s) = B_1 \wedge \dots \wedge B_n$ を考える. いま, 拡張ソート $ES(X: s)$ が定義 5.4 の条件 (1) および (2) を満足すると仮定する. この時, KB 中出现しない定数による es の基礎化代入 θ に対し,

$$T \cup \{es\theta\} \vdash_{\mathcal{H}} \exists ES(a) \quad (7)$$

$$T \not\vdash_{\mathcal{H}} \exists ES(a) \quad (8)$$

が成り立つ. ここで, $X\theta = a$ である.

《証明》

式 (7) について: 条件 (1) を満足することから明らか. 式 (8) について: $T \vdash_{\mathcal{H}} \exists ES(a)$ であると仮定する. この時, $T \vdash_{\mathcal{H}} ES\sigma$ かつ $X: \sigma = a$ となる ES の基礎化代入 σ を考えることができる.

ここで, $ES\sigma$ について, KB 中出现しない定数をもとの変数に戻した結果を $\forall ES'$ で表すと, $T \vdash_{\mathcal{H}} \forall ES'$ が成り立つ. この時, ES のルート変数 X に代入された a は, KB 中出现しない定数であるから, $\forall ES'$ には X が出現するはずである. このことは, ES をゴールとした時, X に何も代入しない解代入が存在することを意味するから, 条件 (2) を満たすことと矛盾する. よって, 式 (8) が成立することがわかる. □

候補生成の観点からみると, 定理 5.2 より次のことが明らかとなる.

候補は, $T \cup \{es\theta\}$ からの前向き推論で得られるもののうち, 特に, T に $es\theta$ を追加することではじめて導出されるものだけを考えれば十分である.

すなわち, 考慮すべき候補は, $T \cup \{es\theta\}$ から導出され, かつ, その構成要素として, T のみからは導出できないアトムを少なくともひとつ含むものだけで十分である. こうした候補の生成は, SOL-導出を利用して実現することができる.

与えられた一階述語論理における節集合 Σ から導出可能な節のうち, 生成領域 $\mathcal{P}$ に記述された制約 (条件) を満たすものを, Σ の $\mathcal{P}$ に関する特徴節 (Characteristic Clauses) と呼び, それらの集合を $Carc(\Sigma, \mathcal{P})$ で表すことにする. SOL-導出 (Skipping Ordered Linear Reso-

lution) [Inoue 92] は, $Carc(\Sigma, \mathcal{P})$ を求めることを目的に提案された順序付き線形導出法である.

いま, ある節 C について,

$$NewCarc(\Sigma, C, \mathcal{P}) = \\ Carc(\Sigma \cup \{C\}, \mathcal{P}) - Carc(\Sigma, \mathcal{P})$$

を考える. $NewCarc(\Sigma, C, \mathcal{P})$ は, $\Sigma \cup \{C\}$ から導出される特徴節のうち, C の追加が原因で得られるものだけからなる. ここで, $NewCarc(\Sigma, C, \mathcal{P})$ 中の各節は, $\Sigma, \mathcal{P}$ に関する C の新特徴節と呼ばれ, SOL-導出は, こうした新特徴節の導出に関して完全な導出法となっている. すなわち, $NewCarc(\Sigma, C, \mathcal{P})$ 中の各節は, $\Sigma \cup \{C\}$ からの SOL-導出により必ず求めることができる. さらに, そこでは, 追加される節 C を先頭節とする導出が行なわれるため, 新特徴節でない節 (つまり, Σ のみから導出される節) の生成は, 少なく抑えられる. なお, 詳細については割愛するが, 節集合 Σ の特徴節集合 $Carc(\Sigma, \mathcal{P})$ の計算は, Σ 中の各節の新特徴節集合の計算に帰着できることから, SOL-導出は $Carc(\Sigma, \mathcal{P})$ の計算に関しても完全 (かつ健全) であることが示されている [Inoue 92]. それに従い, SOL-導出は, $Carc(\Sigma, \mathcal{P})$ の計算を目的として用いられるのが一般的であることを付け加えておく.

導出過程で用いる代入を, 順序ソート代入に置き換えることで, 文献 [Inoue 92] での一階述語論理における議論は, 順序ソート論理においても同様に展開できる [森田 99]. 以下では, 順序ソート論理における SOL-導出のもとで議論を続ける.

先の定理 5.2 の性質に基づく候補生成処理に戻ろう. まず, $\mathcal{T} \cup \{es\theta\}$ から導出可能なアトム集合は $Carc(\mathcal{T} \cup \{es\theta\}, \mathcal{P})$ と表すことができ, SOL-導出を用いて同定することができる. ここで, 生成領域は

$$\mathcal{P} = \langle \text{単一アトムからなる節} \rangle$$

である. こうしたアトム集合の中で, $\mathcal{T}$ から導出できないもの, すなわち, $\mathcal{T}$ に $es\theta$ を追加したことではじめて導出されるアトムを同定する必要があるが, $es\theta$ はアトムの連言, つまり, 節の連言 (節集合) となることに注意しよう. SOL-導出は, ある単一節の新特徴節の導出に関して完全な導出法であることは先に述べたが, こうした計算を再帰的に繰り返すことで, 節集合の追加により初めて導出される特徴節についても, 完全であることが示されている [Inoue 92, 森田 99]. こうした特徴節を, 節集合の新特徴節と定め, 先と同様な表記を用いることにする. すなわち,

$$NewCarc(\mathcal{T}, es\theta, \mathcal{P}) = \\ Carc(\mathcal{T} \cup es\theta, \mathcal{P}) - Carc(\mathcal{T}, \mathcal{P})$$

を, $\mathcal{T}, \mathcal{P}$ に関する $es\theta$ の新特徴節集合とする. ここでも単一節の場合と同様, 追加される節それぞれを先頭節とすることで, $es\theta$ の新特徴節でない節の生成を少なく抑えながら, 新特徴節をすべて導出することが可能であ

る. つまり, $\mathcal{T} \cup es\theta$ からの SOL-導出により実際に得られる節 (アトム) 集合を $Prod(\mathcal{T}, es\theta, \mathcal{P})$ とすると,

$$NewCarc(\mathcal{T}, es\theta, \mathcal{P}) \subseteq Prod(\mathcal{T}, es\theta, \mathcal{P}) \quad (9)$$

が成り立つ.

SOL-導出を用いて得られる $Prod(\mathcal{T}, es\theta, \mathcal{P})$ と $Carc(\mathcal{T} \cup es\theta, \mathcal{P})$ をもとに, 定理 5.2 に基づく特殊化拡張ソートの候補生成を次の通り行なう.

$ES' \subseteq Carc(\mathcal{T} \cup es\theta, \mathcal{P})$ なる ES' のうち, a が出現するアトムを含み, かつ, $ES' \cap Prod(\mathcal{T}, es\theta, \mathcal{P}) \neq \emptyset$ であるものを取り出す. ここで, es のルート変数 X に対して $X\theta = a$ である. こうした ES' に対して, $ES\sigma = ES'$ となる拡張ソート ES を候補とする. ただし, $X\theta = X\sigma = a$ であり, θ 中に用いられた定数は ES に出現しないものとする.

厳密に述べると, ここで得られる候補 ES は, 式 (7) を満たすものではあるが, 必ずしも式 (8) を満たすわけではない. なぜなら, 式 (9) が成り立つことから, 一般に, $Prod(\mathcal{T}, es\theta, \mathcal{P})$ は $\mathcal{T}$ から導出可能なアトムも含むからである. しかし, $Prod$ の計算手続きにより, こうしたアトムは極力少なく抑えられており, それに伴い, 式 (8) を満たさない候補も多数とはならないことが期待できる. よって, 上述の手順に従い, 不要な候補生成の枝刈りが実現できると言えよう.

§2 候補の検査

前節の手順により得られた候補は, 定義 5.4 の条件 (2), (3) および (4) を満足するか否かを検査され, すべてを満たすものが求めたい特殊化拡張ソートとなる^{*10}. 条件 (2) と (4) の検査は, 具体的には次の通り行なわれる. なお, 条件 (3) については説明するまでもない.

条件 (2) の検査:

条件 (2) は, $\mathcal{KB}$ のもとで候補 ES をゴールとした時の任意の計算解代入が, ルート変数 X に何らかの項を代入するものであることを要請する. このことを確かめるには, $\mathcal{KB}$ に出現しない定数 a を代入した $ES(a)$ をゴールとし, $\mathcal{KB}$ のもとでその証明を試みればよい. もし, ES が条件 (2) を満たしていれば, 既に X に定数が代入された $ES(a)$ は証明不可能となるはずである. 逆に, 証明可能であれば, ES をゴールとした際に X へ何も代入しない計算解代入が存在することになり, すなわち, 条件を満たさないことがわかる.

条件 (4) の検査:

修正対象となる節のヘッドを H とすると, 条件 (4) は候補 ES に対して, $ES_0 \subseteq ES$ なる任意の ES_0 について $H \not\subseteq_{\mathcal{KB}} ES_0$ が成立することを要請する. このことに関して, 以下の定理が成り立つ.

^{*10} 条件 (2) に関する検査の必要性は, 定理 5.2 の逆が一般には成り立たないことによる. 式 (7), (8) の条件を満たす候補のみを考えた場合でも, この検査は省略できないことに注意する.

[定理 5.3]

次のふたつは同値である .

- (1) ES が条件 (4) を満足しない .
- (2) $H \preceq_{KB} A$ なる $A \in ES$ が存在する .

《証明》

(1) $\Rightarrow$ (2): (1) より, $H \preceq_{KB} ES_0$ なる $ES_0 (\subseteq ES)$ が存在する . その際に考えた H および ES_0 のルート変数を V とする . ここで, V を含む $A \in ES_0$ なるアトム A を考える . 拡張ソートに関する定義より, $ES_0 \preceq_{KB} A$ である . よって, $H \preceq_{KB} A$ が成り立つ .

(1) $\Leftarrow$ (2): $\{A\} \subseteq ES$ であることから明らか . $\square$

いま, $Prod(T, es\theta, P)$ を求めた際に用いた θ を考える . KB に出現しない $H\theta$ 中の定数 c を含む $A' \in Prod(T, es\theta, P)$ なるアトム A' を考える . ここで, $V/c \in \theta$ であるとする . この時, A' が $T \cup \{H\theta\}$ から証明可能であるなら, $A' = A\sigma$ (ただし, A には V が出現し, $V\sigma = c$ である) なる任意の A について, $H \preceq_{KB} A$ が成り立つ . つまり, A' からは, 条件 (4) を満たす拡張ソートの構成アトムは決して得られない . よって候補生成の際にこうした A' は不必要となる . 逆に, こうした A' を, $Prod(T, es\theta, P)$ からすべて除去した上で候補生成を行えば, それら候補は必ず条件 (4) を満たすので, 検査を省略することが可能となる .

5.4 新ソート導入に伴う知識ベースの修正

修正対象となる型付き変数 $X:s$ をボディに含む節 $C = H \leftarrow B_1 \wedge \dots \wedge B_n$ に対して, 拡張ソート $es(X:s) = B_1 \wedge \dots \wedge B_n$ を考えた時, 前節までの議論で, s による型付けを特殊化の際に用いるソート s_α が, es に関する s の特殊化拡張ソート $ES(X:s)$ で定義されることを述べた . ここでは, こうした新たなソート s_α の導入に伴う領域理論 T , ソート階層 $\mathcal{H}$, および型宣言 $\mathcal{D}$ の具体的な修正処理について述べる .

なお, ES の構成アトムを参照しながら, ユーザは必要に応じて s_α に適当な概念名を与えればよい .

§1 領域理論の修正

式 E における項 t のすべての出現を, 項 t' に置き換えて得られる式を $E_{(t,t')}$ とする . 領域理論の修正にあたっては, T 中の節 C における型付き変数 $X:s$ を, s_α を用いて特殊化すればよい . つまり, C を $C_{(X:s, X:s_\alpha)}$ に置き換えるだけで修正は完了する .

§2 ソート階層の修正

新たなソート s_α が領域理論に導入されたことに伴い, ソート階層の修正を行う . 具体的には, s_α を, もとのソート階層 $\mathcal{H} = (S, \preceq)$ の適切な位置に挿入する . s_α が挿入される位置を同定するには, S 中のソートのうち, s_α を包摂するもの, および s_α に包摂されるものを明らかにする必要がある . これまでの議論から, ソート s が s_α を包摂することは容易にわかる . よって, s_α は s の下位ソートとして挿入される . s_α に包摂されるソート

は, 次の定理から明らかとなる .

[定理 5.4]

知識ベース KB のもとで, 拡張ソート $es(X:s)$ をゴールとした時, $X\theta = Y:t$ (Y は変数) なる計算解代入 θ が存在するならば, $A_t \subseteq E_{KB}(es)$ である .

《証明》

任意の $c \in A_t$ について, c は X に代入可能であるから, $c \in A_s$ かつ $T \vdash_{\mathcal{H}} \exists es(c)$ が成り立つ . よって, $c \in E_{KB}(es)$ である . $\square$

いま, 特殊化拡張ソート ES をゴールとした時の計算解代入のうち, $X:s/Y:t$ を含むものを考え, こうした t の集合を $Lower(ES)$ とする . 定理 5.4 は, s_α が $Lower(ES)$ 中のすべてのソートの上位ソートであることを示している . 特に, 計算解代入は極大なものであることから, $Lower(ES)$ 中のソートとそれらの下位ソート以外が s_α の下位に位置することはない . よって, $Lower(ES)$ 中のソートのみが, s_α の直接の下位ソートとなることがわかる .

§3 型宣言の修正

領域理論・ソート階層の修正に伴い, 定数の型宣言の修正も必要となる . 新ソート s_α は, 特殊化拡張ソート $ES(X:s)$ により定義されるので, s_α の外延は ES の外延と等しくなければならない . 型宣言の修正は, この条件を満たす様に行われる .

ES の外延に含まれる個体は, 以下の定理に従って同定することができる .

[定理 5.5]

KB を知識ベース, $es(X:s)$ を拡張ソートとする . この時, 次のふたつは同値である .

- (1) $c \in E_{M_{KB}}(es)$.
- (2) KB のもとで es をゴールとした時, $X\theta = c$, あるいは, $X\theta = Y:t$ (Y は変数) で $c \in A_t$ となる計算解代入 θ が存在する .

《証明》

(1) $\Rightarrow$ (2): 拡張ソートの外延の定義から, $T \vdash_{\mathcal{H}} \exists es(c)$ である . よって, KB のもとで $es(X:s)$ をゴールとした時の計算解代入のうち, $X\theta = c$, あるいは, $X\theta = Y:t$ (Y は変数) で $c \in A_t$ となる θ が必ず存在する .

(1) $\Leftarrow$ (2): $T \vdash_{\mathcal{H}} es\theta$ である . ここで, c は X に代入可能であるから, $c \in A_s$ かつ $T \vdash_{\mathcal{H}} \exists es(c)$ が成り立つ . よって, $c \in E_{M_{KB}}(es)$ である . $\square$

定理 5.5 から, KB のもとで ES をゴールとした時の計算解代入をすべて求めてやれば, ES に属する個体, つまり, s_α に属する個体が明らかとなる .

定理 5.5 のうち, $X\theta = Y:t$ なる解代入は, t に属する任意の個体が s_α にも属することを意味するが, この場合は, 定理 5.4 に従ってソート階層が修正された結果, t が s_α の下位ソートとなっているはずである . すなわち, t に属する個体は必然的に s_α に属することになる . よって, $X\theta = Y:t$ なる解代入に関して型宣言を修正す

る必要はない。

これに対し、 $X\theta = c$ なる解代入については次に述べる修正が必要となる。解代入 $X\theta = c$ から、個体 (定数) c は s_α に属さなければならないことがわかる。ここで、もとの型宣言における c の主型を u としよう。こうした u の可能性として、 $u = s$ 、および、 $u \neq s$ のふた通りが考えられる。

$u = s$ の場合： c の主型を u から s_α に修正する。 s_α は s の下位ソートであるから、修正後においても c は $u (= s)$ に属することがわかる。よって、この修正がもとの型宣言と矛盾することはない。

$u \neq s$ の場合： c は、 u にも s_α にも属する必要があるがその主型はユニークでなければならない。よって、ソート階層中に、 u と s_α 両者の下位ソートとなる $s_{u\alpha}$ を新たに導入し、 c の主型を u から $s_{u\alpha}$ に修正する^{*11}。その結果、 c は、 u にも s_α にも属することになり、もとの型宣言と矛盾することなしに修正の目的を達成することができる。

ES をゴールとした時の計算解代入それぞれについて、上の処理を行えば型宣言の修正は完了である。

図 1 に、新ソート s_α の導入に伴うソート階層および型宣言の修正例を示す。ここで、 s_α は、拡張ソート $ES(X:s)$ で定義されるものとし、 ES をゴールとした時、 $X\theta_1 = Y:s_3$ 、 $X\theta_2 = Y:s_4$ 、 $X\theta_3 = b$ なる計算解代入が得られたとする。なお、図 1 において、ソート階層中の $s_{c_j}^{s_i}$ なるノードは、個体 (定数) c_j の主型が s_i であることを意味する。 θ_1 および θ_2 から、 s_α は、 s の下位ソートであり、かつ、 s_3 と s_4 の上位ソートであることがわかる (図 1 (2))。また、 θ_3 から、 b は s_α に属さなければならないが、その主型は s_2 である。よって、 s_2 と s_α の下位ソート $s_{2\alpha}$ をさらに導入し、 b の主型を修正する (図 1 (3))。

5.5 修正前後における知識ベースの外延的等価性

これまでの議論に従って知識ベース $\mathcal{KB}$ を修正した結果を $\mathcal{KB}'$ とする。次の定理が示す通り、もとの知識ベースの論理的意味 (外延の意味) は修正前後において保存される。

[定理 5.6]

$\mathcal{KB}$ と $\mathcal{KB}'$ の最小モデルは一致する。

《証明》

修正対象となった節を $C = H \leftarrow B_1 \wedge \cdots \wedge B_n$ とする。これまでの議論に従った修正により、 C 中の s による型付けが、 s_α による型付けに特殊化されている。ここで、 $es(X:s) = B_1 \wedge \cdots \wedge B_n$ とした時、 s_α は、 $es(X:s) \preceq_{\mathcal{KB}} ES(X:s)$ なる拡張ソート $ES(X:s)$ により定義されるソートであり、 s_α の外延 A_{s_α} は、 $ES(X:s)$

の外延 $E_{\mathcal{KB}}(ES)$ と一致する。

いま、 es は C のボディに対応することから、その外延 $E_{\mathcal{KB}}(es)$ は、推論過程で C を用いる際に X へ代入可能な (s に属する) 個体集合を定めていることに注意しよう。 ES は es を包摂するから、 $E_{\mathcal{KB}}(es) \subseteq E_{\mathcal{KB}}(ES)$ であり、すなわち、 $E_{\mathcal{KB}}(es) \subseteq A_{s_\alpha}$ となる。つまり、 C 中の型付き変数 $X:s$ が $X:s_\alpha$ に特殊化されたことにより、 X の取り得る値は A_{s_α} 中の個体のみに限定されるが、そこには、 C を用いる際に代入可能な個体はすべて含まれている。よって、こうした特殊化操作が、推論過程で C が用いられる (発火する) 様子に影響を及ぼすことはない。すなわち、修正の前後で、知識ベースの最小モデルが変化することはあり得ない。□

この様に、修正処理がもとの知識ベースの外延の意味に一切影響を与えないことは、本手法の大きな特徴のひとつである。

5.6 例 題

ここでは、図 2 に示す知識ベース $\mathcal{KB} = \langle \mathcal{H}, \mathcal{D}, \mathcal{T} \rangle$ を具体例として、内包的エラーの修正処理をみていく。

いま、ゴール $G = \text{throw_away_on_friday}(X)$ を考えよう。 G を証明する際に用いる $\mathcal{T}$ 中の節集合 $\mathcal{T}(G)$ は、

```
{ throw_away_on_friday(X) ← incombustible(X),
  incombustible(X:cup) ← made_of(X:cup,Y:metal),
  made_of(X:cup,aluminum:metal) }
```

となり、この時 GDA は cup と can の間に類似性を検出する。これに対し、ユーザはこの類似性を望ましくないものとし、それは cup による変数の片付けが一般的過ぎることに起因すると考えたと仮定しよう^{*12}。

まず、 cup により型付けされた変数をボディに含む $\mathcal{T}(G)$ 中の節

$C = \text{incombustible}(X:\text{cup}) \leftarrow \text{made_of}(X:\text{cup},Y:\text{metal})$ を取り出す。ここで、ボディ $\text{made_of}(X:\text{cup},Y:\text{metal})$ を、ルート変数が $X:\text{cup}$ である拡張ソートと考え、これを es とする。定理 5.2 に従って、もとの知識ベースに出現しない定数 c_1 と c_2 を用いた代入 $\theta = \{X:\text{cup}/c_1, Y:\text{metal}/c_2\}$ による es の基礎例 $es\theta = \text{made_of}(c_1, c_2)$ を考え、 cup の特殊化拡張ソートの候補を生成する。ここでは、候補の一例として $\text{has_handle}(X:\text{cup})$ を考え、この候補が特殊化拡張ソートの条件 (2),(3),(4) を満足するか否かを検査する。知識ベースに出現しない定数 c_3 をルート変数に代入した $\text{has_handle}(c_3)$ は、 $\mathcal{KB}$ のもとで証明不可能である。よって、条件 (2) を満足することがわかる。また、条件 (3) を満たすことは明らかである。

*11 u が s_α の下位に位置するなら、 $s_{u\alpha}$ の導入は必要ないと考えられるが、計算解代入は極大なものであることから、今の場合、それはあり得ない。

*12 例えば、このユーザは cup に対して、デフォルトで紙性のものを連想する場合が考えられよう。

図 1 新ソートの導入に伴うソート階層および型宣言の修正

領域理論 T :

```

throw_away_on_friday(X) ← incombustible(X).
throw_away_on_tuesday(X) ← combustibile(X).
incombustible(X:cup) ← made_of(X,Y:metal).
incombustible(X:can) ← made_of(X,Y:metal).
has_handle(X:cup) ← made_of(X,Y:metal).
made_of(X:aluminum_cup,aluminum).
made_of(X:plastic_cup,plastic).
made_of(X:ceramic_cup,ceramic).
made_of(X:steel_can,steel).
made_of(X:aluminum_can,aluminum).

```

ソート階層 $\mathcal{H}$ ・型宣言 $\mathcal{D}$:

図 2 知識ベース

さらに, C のヘッド $\text{incombustible}(X:\text{cup})$ に対し, $\text{incombustible}(X:\text{cup}) \not\vdash_{KB} \text{has_handle}(X:\text{cup})$ であるから, 定理 5.3 より, 条件 (4) を満たすことも確認できる. 以上から, $\text{has_handle}(X:\text{cup})$ は, cup の特殊化拡張ソートであることがわかる. よって, C 中の型付き変数 $X:\text{cup}$ を特殊化の際に用いるソート s_α は, $\text{has_handle}(X:\text{cup})$ により定義されるものとする. ここで, s_α に具体的な概念名を付与する必要がある場合は, その定義に基づいて, 例えば『取って付きカップ』などの概念名を与えればよいだろう.

こうしたソート s_α を用いて節 C を $C_{(X:\text{cup}, X:s_\alpha)}$ とすなわち,

$$\text{incombustible}(X:s_\alpha) \leftarrow \text{made_of}(X,Y:\text{metal})$$

に置き換えることで領域理論の修正は完了である.

次に, s_α をもとのソート階層 $\mathcal{H}$ 中の適切な位置に挿入しよう. s_α は cup の下位に位置することは明らかである. また, s_α を定義した特殊化拡張ソート $\text{has_handle}(X:\text{cup})$ をゴールとして KB のもとで証明を試みると,

$$\text{Lower}(\text{has_handle}(X:\text{cup})) = \{\text{aluminum_cup}\}$$

が得られ, すなわち, aluminum_cup のみが s_α の下位ソートとなることがわかる. これに基づき修正を行なうと, 図 3 に示すソート階層が得られる.

なお, この例においては, 型宣言の修正は必要ない.

図 3 修正後のソート階層 (修正部分のみ)

6. 実験

本章では, 提案した知識修正アルゴリズムの実装システムによる実験結果を示す. なお, システムは SICStus Prolog で実装し, SUN SparcStation-5 上で実行した.

実験を行なうにあたり, 2 種類の知識ベースを用意した. サイズ ([領域理論中の節数, ソート階層中のソート数]) は, それぞれ [10, 9] と [40, 16] である. 前者は図 2 の知識ベースであり, 後者はそれに節や階層を追加して作成した. なお, 以下実験結果では, 各知識ベースをそのサイズで参照している.

本知識修正手法において最もコストが掛かるのは, 型付き変数を特殊化するために用いる特殊化拡張ソートの計算である. その効率は, 特殊化拡張ソートの候補生成処理に強く依存することは既に述べた. $KB = \langle \mathcal{H}, \mathcal{D}, T \rangle$ を知識ベース, $C = H \leftarrow B_1 \wedge \dots \wedge B_n$ を特殊化の対象となる型付き変数を含む節とした時, $es = B_1 \wedge \dots \wedge B_n$

表 1 実験結果 1 (知識ベースサイズ: [10, 9])

候補生成手法	導出されたアトム数	計算時間 (msec.)
FWR	4	80
SOL	4	10

表 2 実験結果 2 (知識ベースサイズ: [40, 16])

候補生成手法	導出されたアトム数	計算時間 (msec.)
FWR	6	1,130
SOL	6	40

とすると、前章での議論から、次の 2 通りの候補生成手法が考えられる。

- (1) $\mathcal{T} \cup \{es\theta\}$ からの前向き推論による候補生成。
- (2) (1) で生成される候補のうち、 $\mathcal{T}$ からの前向き推論で得られるものは除外する SOL-導出を用いた候補生成。

これらの手法をそれぞれ、FWR, SOL で参照することにしよう。ここでは、それぞれの候補生成手法を用いて特殊化拡張ソートを求めた際の結果を比較・考察することで、SOL-導出を用いることの効果を確認する。なお、各知識ベースに対して、それぞれひとつの特殊化拡張ソートが得られ、各々は単一のアトムからなるものであった。また、これら特殊化拡張ソートの導入に伴うソート階層および型宣言の修正処理は、どちらの場合も 10 msec. の時間を要した。

実験結果を表 1・表 2 に示す。

実験結果から、SOL-導出を用いることで、計算時間が大きく改善されたことがわかる。特に、知識ベースサイズが大きい場合に、その効果がより著しいことが認められ、知識ベースが大規模になるほど、より大幅な処理効率改善が期待できると思われる。なお、本実験では、導出されたアトム数はいずれの結果においても FWR, SOL とともに同じであったが、一般には SOL によりその数は減少し、それらをもとに生成される候補数も結果として少なくなる^{*13}。以上のことから、定理 5.2 の与える性質に基づき、特殊化拡張ソートの候補生成処理に SOL-導出を用いたことが、効率化を実現する上で極めて有効であることが確認できたと言える。

7. お わ り に

本稿では、知識ベースに反映された類似性の不具合をトリガーとする、従来とは異なる新たな知識修正手法を提案し、その実現アルゴリズムを示した。そこでは、これまでの外延的アプローチでは扱うことが出来ない内包的エラーのひとつである知識の一般的過ぎる型付けを、特殊化拡張ソートを計算することにより適切な型付けに修正する。特に、こうした修正の前後において、もとの知識の論理的意味が保存される点が特徴である。外延的エ

ラーを修正する従来のシステムと本内包的エラー修正システムを統合することで、エラーを修正するために十分な正例・負例を具体的かつ容易に与えることができる場合は、従来の外延的アプローチで対処し、(外延的には観測できない) 型付けに関するエラーを修正する際には本手法で対処するといった、異なるタイプのエラーを扱うことのできる柔軟かつ強力なハイブリッド知識修正システムの構築が可能となろう。

今後の重要な課題として、一般的過ぎる型付けが複数ある場合の修正処理の考察が挙げられる。これらの修正の順序は、修正後の知識ベース、および、そこに反映される類似性に大きく影響を与える。よって、おののおを独立に扱う単純な処理では不十分であり、さらなる考察が必要である。

また、本稿での知識修正のトリガーは、ユーザにとって望ましくない類似性の観測であったが、逆に、望ましくない非類似性の観測をトリガーとする手法も考えられる。つまり、ユーザは、ある概念同士を類似していると考えているにも関わらず、非類似な概念として観測される場合である。このことは、それら概念間で共有されるべき内包的性質が欠落していることを意味し、知識の補完処理が修正の核となろう。こうした処理を定式化し、本稿での枠組と統合することも大変重要な課題である。

謝 辞

有益なコメントを頂いた査読者の方々に深く感謝致します。なお、本研究は一部、文部省科学研究費基盤研究 (C)(2) 課題番号 11680375、および、同特定領域研究 (A)(1) 課題番号 10143104 の補助を受けた。

◇ 参 考 文 献 ◇

- [Falkenhainer 89] B. Falkenhainer, K. D. Forbus and D. Gentner: The Structure Mapping Engine: Algorithm and Examples, *Artificial Intelligence*, 41(1), pp. 1–63, 1989.
- [Gentner 83] D. Gentner: Structure Mapping: A Theoretical Framework for Analogy, *Cognitive Science*, 7, pp. 155–170, 1983.
- [Goldstone 91] R. L. Goldstone, D. L. Medin and D. Gentner: Relational Similarity and the Nonindependence of Features in Similarity Judgments, *Cognitive Psychology*, 23, pp. 222–262, 1991.
- [Haraguchi 00] M. Haraguchi, Y. Okubo and T. Kakuta: A Logical Foundation of Metaphoric Search Based on Abstraction Approach, *Electronic Transactions on Artificial Intelligence*, 4(Section B), pp. 53–81, 2000.
- [原口 99] 原口 誠・森田 展博: 知識の修正・更新問題への内包的アプローチ, 「発見科学」平成 11 年度 A02 班第 2 回班会議報告資料, 1999.
- [Inoue 92] K. Inoue: Linear Resolution for Consequence-Finding, *Artificial Intelligence*, 56(2–3), pp. 301–353, 1992.
- [Kakuta 97] T. Kakuta, M. Haraguchi and Y. Okubo: A Goal-Dependent Abstraction for Legal Reasoning by Analogy, *Artificial Intelligence & Law*, 5(1–2), pp. 97–118, 1997.
- [Lifschitz 01] V. Lifschitz, D. Pearce and A. Valverde: Strongly Equivalent Logic Programs, *ACM Transactions on Computational Logic*, 2(4), pp. 526–541, 2001.

*13 最悪の場合でも FWR によるそれを上回ることはない。

- [森田 99] 森田 展博: 型の特殊化による類似性駆動型知識修正法について, 北海道大学 大学院工学研究科 修士論文, 1999.
- [Morita 99] N. Morita, M. Haraguchi and Y. Okubo: A Method of Similarity-Driven Knowledge Revision for Type Specializations, Proc. of the 10th Intl. Conf. on Algorithmic Learning Theory – ALT'99, LNAI 1720, pp. 194–205, 1999.
- [Okubo 98] Y. Okubo and M. Haraguchi: Constructing Predicate Mappings for Goal-Dependent Abstraction, Annals of Mathematics and Artificial Intelligence, 23, pp. 169–197, 1998.
- [Tenenbergs 89] J. D. Tenenbergs: Abstracting First-Order Theories, Change of Representation and Inductive Bias (P. D. Benjamins, ed.), Kluwer Academic Publishers, pp. 67–79, 1989.
- [Walter 88] C. Walter: Many-Sorted Unification, Journal of the Association for Computing Machinery, 35(1), pp.1–17, 1988.
- [Wrobel 94] S. Wrobel: Concept Formation and Knowledge Revision, Kluwer Academic Publishers, Netherlands, 1994.

〔担当委員: 新田克己〕

2002 年 2 月 5 日 受理

◇ 付 録 ◇

A. 特殊化拡張ソートの満たすべき条件

定義 5.4 における条件 (2) の必要性は次の定理に述べられる.

〔定理 A.1〕 $X:s$ をルート変数とする拡張ソート ES を考える. 知識ベース $\mathcal{KB}$ において, ES をゴールとした時の証明を考える. この時, 次のふたつは同値である.

- (1) 条件 (2) を満たさない, すなわち, $X\theta = X$ なる計算解代入 θ が存在する.
- (2) $E_{\mathcal{KB}}(ES) = A_s$.

《証明》

(1) $\Rightarrow$ (2): $E_{\mathcal{KB}}(ES) \subseteq A_s$ は明らか.

ES をゴールとした時に $X\theta = X$ なる計算解代入が存在するのであるから, $\mathcal{T} \vdash_{\mathcal{H}} \forall X:s \exists ES(X)$ が成り立つ. よって, 任意の $c \in A_s$ について, $c \in E_{\mathcal{KB}}(ES)$ である.

(1) $\Leftarrow$ (2): $E_{\mathcal{KB}}(ES) = A_s$ であるから, 定義 5.2 より, $\mathcal{T} \vdash_{\mathcal{H}} \forall X:s \exists ES(X)$ が成り立つ. よって, ES をゴールとした時, $X\theta = X$ なる解代入 θ が存在する. $\square$

以上から, ゴール ES に対する任意の計算解代入が X への代入を行うものであるなら, $A_s \neq E_{\mathcal{KB}}(ES)$ であり, よって, $E_{\mathcal{KB}}(ES) \subset A_s$ となる. すなわち, ES は s の真の下位ソートに相当することがわかる.

条件 (3) と (4) の必要性は次の通りである.

〔定理 A.2〕 ES が条件 (3) あるいは (4) を満足しない時, 修正後の領域理論には冗長な記述が含まれる.

《証明》 修正対象となる節 $H \leftarrow es$ (ここで $es = B_1 \wedge \dots \wedge B_n$) の型付き変数 $X:s$ を, 特殊化拡張ソート ES により定義される新ソート s_α を用いて特殊化すると, $H_{(X:s, X:s_\alpha)} \leftarrow es_{(X:s, X:s_\alpha)}$ となる. この式は, s_α をその定義に基づいて展開することで

$$H \leftarrow es \wedge ES \quad (\text{A.1})$$

と表すことができるが, ES が条件 (3) あるいは (4) を満たさない場合は, そこに記述の冗長性が生じることを示す.

条件 (3) を満たさない場合, つまり, $es \cap ES \neq \phi$ の場合, 式 (A.1) のボディには同じアトムが重複して出現する. よって, こうした記述は冗長であると言える.

条件 (4) を満たさない場合, つまり, $H \preceq_{\mathcal{KB}} ES_0$ なる $ES_0 \subseteq ES$ が存在する場合を考える.

θ を $\mathcal{KB}$ に出現しない定数による H の基礎化代入とすると, これまでの議論から, $\mathcal{T} \vdash_{\mathcal{H}} ES_0\sigma \leftarrow H\theta$ なる ES_0 の基礎化代入 σ が存在する. ここで, $\mathcal{KB}$ に出現しない $ES_0\sigma$ 中の定数をもとの変

数に戻した結果を ES'_0 とし, $ES_0\sigma' = ES'_0$ なる代入 σ' を考えると, $\mathcal{T} \vdash_{\mathcal{H}} ES_0\sigma' \leftarrow H$ が成り立つ. $ES_0\sigma' \leftarrow H$ において, H は条件, ES_0 は結論として現れるのに対し, 式 (A.1) では, ES_0 は条件, H は結論として現れる. これは, 修正後の知識ベースでは, ある条件からそれと同じことが再度結論づけられることを意味する. よって, こうした記述は冗長であると言える. $\square$

—— 著 者 紹 介 ——

大久保 好章(正会員)

1990 年 千葉大学工学部機械工学科卒業. 1992 年 東京工業大学大学院総合理工学研究科システム科学専攻修士前期課程修了. 1995 年 同博士後期課程修了, 博士(理学). 1995 年 北海道大学工学部助手. 現在, 同大学大学院工学研究科助手. 機械学習, 知識獲得の研究に従事. AAAI, 情報処理学会, 日本ソフトウェア科学会各会員.

森田 展博

1997 年 北海道大学工学部電気工学科卒業. 1999 年 同大学大学院工学研究科電子情報工学専攻修士課程修了. 在学中は, 知識獲得・修正手法の研究に従事.

原口 誠(正会員)

1976 年 九州大学理学部数学科卒業. 1978 年 同大学大学院数学専攻修士課程修了. 1979 年 鹿児島大学理学部数学教室助手. 1981 年 九州大学理学部付属基礎情報学研究施設助手. 1984 年 理学博士(九州大学). 1987 年 東京工業大学大学院総合理工学研究科助教授. 1995 年 北海道大学工学部教授. 現在, 同大学大学院工学研究科教授. データマイニング, 知識表現システム, 演繹データベース, 機械学習の研究に従事. 情報処理学会, 日本ソフトウェア科学会各会員.