

# 単一例題からの帰納的学習のための類推的一般化

大川 剛直<sup>†</sup> 馬場 口 登<sup>††</sup>

新しい知識を獲得するための有力な方法として帰納的学習が挙げられるが、これは通常、多数の例題を必要とする。一方、人間はしばしば既存の知識を利用することにより、与えられた例題が少数であった場合にも適切な一般化を行い、新しい妥当な知識を導き出すことが可能である。本論文では、帰納的学習の枠組で、既存の類似ルールをガイドとして単一の例題を一般化する類推的一般化と呼ぶ新しい手法を提案する。例題とルール間の類似性を、各々を構成する原子式の演繹的閉包間の包含関係に基づいて定式化することにより、形式的な操作で類似ルールを選択することが可能である。英文構文解析システムにおけるルール生成実験の結果、類推的一般化により、単一例題から適切なルールが生成されることを確認した。

## Analogical Generalization for Inductive Learning from a Single Example

TAKENAO OHKAWA<sup>†</sup> and NOBORU BABAGUCHI<sup>††</sup>

Learning from examples is widely used in the paradigm of inductive learning. The difficulty in learning from examples is that a number of examples are usually needed to obtain satisfactory learning results. On the other hand, humans might acquire new knowledge only from a few examples. It will be effective to make highly use of the similar known knowledge in generalizing a few examples as humans will do. In this paper, we propose a new inductive learning scheme, called analogical generalization, which is capable of generating a new rule from a single example, using existing rules as a guide. Analogical generalization is based on the similarity between a given example and existing rules, both of which are expressed as logical formulas. We define the similarity as the subsumption relation between deductive closures of atoms forming the examples and the rules. Through the experiment of rule generation in a system for parsing English sentences, we have confirmed that analogical generalization is useful to acquire new valid rules.

### 1. ま え が き

知識獲得ボトルネック解消のための最も有力な方法の一つとして機械学習による知識の自動獲得が挙げられる。中でも、例題をもとに一般的な概念を獲得する帰納的学習 (inductive learning) は概念学習とも呼ばれ、新しい知識の生成が可能であり、また、我々が持つ学習のイメージに近いものとも言えるため、これまでに数多くの方式が提案されている<sup>1),2)</sup>。

帰納的学習における主要な操作は一般化 (generalization) と呼ばれるものである。すなわち、与えられ

た正の例題をすべて満たし、負の例題を一つも満たさないように例題を一般化することにより、一般的概念が獲得される。このとき、適切な一般化を行い、良好な学習結果を得るために、通常、多数の例題を必要とする。

一方、我々人間が新たな知識を獲得する場合、常に多数の例題を必要とするわけではなく、与えられた例題が少数であった場合にも、新しい知識を導き出すことがしばしばある。このような考えのもとに、少数の例題から学習可能な方式がいくつか考案されている。例えば、領域理論と呼ばれる豊富な背景知識を前提とした学習法である説明に基づく学習 (explanation based learning)<sup>3)</sup> や Russell の単一例題一般化 (single instance generalization)<sup>4)</sup> などがその代表的なものといえよう。しかしながら、これらの手法によって得られる知識は、正当性が保証されているものの、既知の知識から演繹的に導かれるものに限られており、新し

<sup>†</sup> 大阪大学工学部情報システム工学科

Department of Information Systems Engineering, Faculty of Engineering, Osaka University

<sup>††</sup> 大阪大学産業科学研究所

Institute of Scientific and Industrial Research, Osaka University

い知識の獲得を目的とする帰納的な学習を実現したものではない。

本研究では、人間が行っているような少数の例題からの帰納的学習を実現するためには、例題の適切な一般化を行う際に自らが既に持っている知識を利用することが重要と考え、類似した既知知識の利用という類推 (analogical reasoning) 的な考え方を帰納的学習に導入した類推的一般化を定式化する<sup>5)</sup>。

英語の初学者が英文法を学習するような状況を考えよう。例えば、簡単な過去形文章の構文規則の学習は、通常の帰納的学習の枠組からとらえると、多数の過去形の例文から共通するエッセンスを抽出し、一般的規則として構成することにはほかならない。しかしながら、仮に学習者が既に幾つかの構文に関する知識を有しているならば、一つか二つの過去形の例文を与えただけで、それに類似した構文である現在形の構文規則を想起して利用することにより、一層効果的に過去形の構文規則を獲得できる。類推的一般化はこのような人間の学習プロセスのモデル化を試みたものである。

類推的一般化は与えられた例題と既知の知識ベースに存在するルールとの間の類似性を評価し、最も類似している既知ルールをガイドとして例題を一般化することにより、単一例題から新しいルールを獲得する。類似性評価の際には、評価対象となる記述が既知の知識のもとでどのような新しい情報を提供するかを反映させた尺度を用いる。その結果、述語記号の一致など表層的な記述の同一性による類似性評価と異なり、あらかじめ類似性を明示する必要がなく、また、相対的な類似性の評価が可能である。以下、類似性の尺度について論じた後、類推的一般化を定式化し、その適用例について述べる。

## 2. 類推的一般化

図1に類推的一般化の概略を示す。その基本的な動作は、単位節の集合として与えられた一つの例題 (example)  $E$  を一般化 (generalization) することにより、ホーン節形式のルール  $r$  を生成するものである。一般化の際には最も例題に類似したルール (ベースルールと呼ぶ)  $r'$  が既知ルール集合  $R$  から選択され、ガイド (guide) として利用される。

このとき問題となるのが類似したルール選択のための類似性 (similarity) の評価尺度である。ここでは、与えられた事実と既存の知識から導かれる新事実との共通点を考慮することにより、事実間の類似性を評価することが可能と考え、演繹的閉包の包含関係に基づいた類似性の定義を与える。以下では、類推的一般化

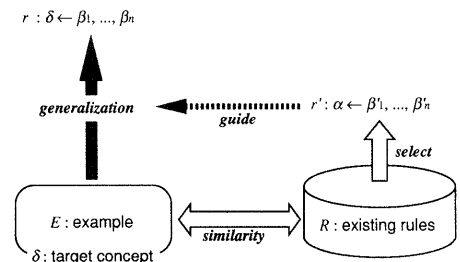


図1 類推的一般化

Fig. 1 Analogical generalization.

の定式化に先立ち、まず原子式間の類似性を定義し、次いで原子式集合間の類似性を定式化する。

### 2.1 原子式間の類似性

最初にいくつかの記法を定義する。代入とは、 $v/t$  の対からなる有限集合である。ただし、 $v$  は変数、 $t$  は項を表す。 $\theta = \{v_1/t_1, \dots, v_n/t_n\}$  を代入、 $e$  を式とすると、 $e\theta$  は  $e$  に出現する変数  $v_i$  をすべて  $t_i$  に置換することにより得られる式を表す。 $S$  が式式の有限集合のとき、 $S\theta$  は集合  $\{e\theta \mid e \in S\}$  を意味する。また、 $S \vdash e$  は式  $e$  が  $S$  から演繹可能なことを意味する。

ルール  $r$  の前提部、帰結部を各々  $prem(r)$ ,  $cons(r)$  と表す。ここで、既存のルール集合  $R$  内のすべてのルール  $r$  に対し、構文上生成的 (syntactically generative)<sup>6)</sup> であるという制限を設ける。すなわち、 $cons(r)$  に出現するすべての変数は  $prem(r)$  に出現するものとする。

以下では二つの関数を与え、これをもとに原子式間の類似性を定義する。なお、以下の定義で  $\alpha, \alpha'$  は原子式を表すものとする。

定義 2.1 ( $\alpha$  の  $R$ -演繹集合  $\Phi(R, \alpha)$ )

$$\Phi(R, \alpha) \stackrel{\text{def}}{=} \{\beta \mid R \cup \{\alpha\} \vdash \beta,$$

$\beta$  は  $\beta \neq \alpha, R \not\vdash \beta$  なる原子式}.  $\square$

定義 2.2 ( $\alpha$  の  $\alpha'$  に対する  $R$ -共通集合  $\Psi(R, \alpha, \alpha')$ )  $\Sigma \theta = \Phi(R, \alpha)$  なる定数を含まない原子式の集合  $\Sigma$ , 要素数最小の代入  $\theta$  に対して、代入  $\theta'$  が存在し、

$$\Sigma' \theta' \subseteq \Phi(R, \alpha')$$

となる  $\Sigma$  の極大部分集合を  $\Sigma'$  とするとき、 $\Sigma' \theta$  を  $\alpha$  の  $\alpha'$  に対する  $R$ -共通集合と呼び、 $\Psi(R, \alpha, \alpha')$  と記す。ただし、 $\alpha \neq \alpha'$ .  $\square$

図2は  $R$ -演繹集合 ( $R$ -deducible set) を図示したものである。例えば、既存の知識集合 (existing knowledge) として「鳥は飛ぶ ( $\text{fly}(x) \leftarrow \text{bird}(x)$ )、鳥は動物である ( $\text{animal}(x) \leftarrow \text{bird}(x)$ )、飛行機は飛ぶ ( $\text{fly}(x) \leftarrow \text{airplane}(x)$ )、飛行機は機械である

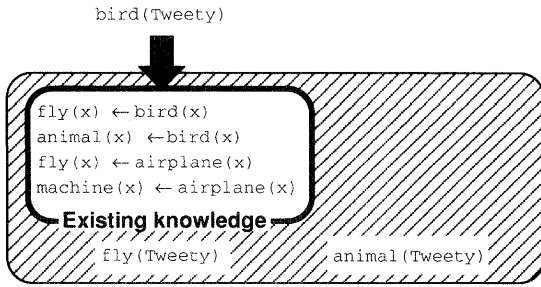


図2 R-演繹集合  
Fig. 2 R-derivable set.

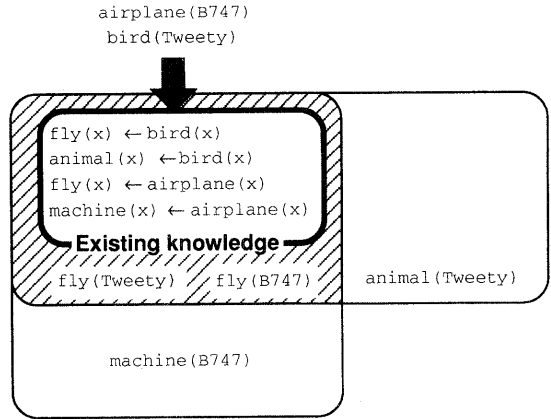


図3 R-共通集合  
Fig. 3 R-common set.

「 $machine(x) \leftarrow airplane(x)$ 」を所持していた時に、「Tweetyは鳥である ( $bird(Tweety)$ )」という新しい情報が与えられることにより、新たに(演繹的に)獲得した斜線部の知識集合「Tweetyは飛ぶ ( $fly(Tweety)$ )、Tweetyは動物である ( $animal(Tweety)$ )」が  $R$ -演繹集合となる。また、 $R$ -共通集合 ( $R$ -common set) は、二つの異なる事実が明らかになったときに共通に得られる知識集合である。例えば同じ既存の知識に対して、「Tweetyは鳥である ( $bird(Tweety)$ )」と「B747は飛行機である ( $airplane(B747)$ )」という二つの事実が与えられた時に得られる共通の特徴である図3の斜線部分「TweetyもB747も飛ぶ ( $fly(Tweety)$ ,  $fly(B747)$ )」から得られる  $fly(Tweety)$  がこれに相当する。ある事実から導出可能な事実の集合により、その事実の性質が表されると考えると、 $R$ -共通集合は二つの事実に共通した性質を表すものであり、背景知識の存在下における事実の間の関連性を反映したものととらえることができる。そこで、より多くの共通した性質を持つ事実を類似したものと考え、 $R$ -共通集合を利用し、類似性を以下のように定義する。

**定義 2.3 (原子式間の類似性)** 既存ルール集合  $R$ , 原子式  $\alpha, \alpha_1, \alpha_2$  に対し、

$$\Psi(R, \alpha, \alpha_2) \subseteq \Psi(R, \alpha, \alpha_1)$$

が成立するとき、原子式  $\alpha$  は  $R$  のもとで\*原子式  $\alpha_2$  より  $\alpha_1$  に類似しているといい、

$$\alpha : \alpha_1 \stackrel{R}{\triangleright} \alpha_2$$

と記す。また、

$$\Psi(R, \alpha, \alpha_2) \not\subseteq \Psi(R, \alpha, \alpha_1)$$

が成立するとき、原子式  $\alpha$  は  $R$  のもとで原子式  $\alpha_2$  より  $\alpha_1$  に真に類似しているといい、

$$\alpha : \alpha_1 \stackrel{R}{\triangleright} \alpha_2$$

と記す。 □

例えば、 $R_1$  を以下に示すルール集合とする。

$$R_1 = \{ \text{parent}(x, y) \leftarrow \text{father}(x, y), \\ \text{parent}(x, y) \leftarrow \text{mother}(x, y), \\ \text{family}(x, y) \leftarrow \text{parent}(x, y), \\ \text{family}(x, y) \leftarrow \text{brother}(x, y), \\ \text{hates}(x, y) \leftarrow \text{kills}(x, y), \\ \text{hates}(x, y) \leftarrow \text{hurts}(x, y), \\ \text{hates}(x, y) \leftarrow \text{strikes}(x, y) \}$$

ここで、原子式  $\text{father}(\text{Lucy}, \text{Jack})$  の  $\text{mother}(\text{Jim}, \text{Betty})$  と  $\text{brother}(\text{Tom}, \text{Joe})$  に対する類似性を考えてみよう。各々の原子式に対して、以下の  $R$ -演繹集合が得られる。

$$\Phi(R_1, \text{father}(\text{Lucy}, \text{Jack})) \\ = \{ \text{parent}(\text{Lucy}, \text{Jack}), \text{family}(\text{Lucy}, \text{Jack}) \} \\ \Phi(R_1, \text{mother}(\text{Jim}, \text{Betty})) \\ = \{ \text{parent}(\text{Jim}, \text{Betty}), \text{family}(\text{Jim}, \text{Betty}) \} \\ \Phi(R_1, \text{brother}(\text{Tom}, \text{Joe})) \\ = \{ \text{family}(\text{Tom}, \text{Joe}) \}.$$

これより、 $\text{father}(\text{Lucy}, \text{Jack})$  の  $\text{mother}(\text{Jim}, \text{Betty})$  に対する  $R$ -共通集合および  $\text{brother}(\text{Tom}, \text{Joe})$  に対する  $R$ -共通集合は

$$\Psi(R_1, \text{father}(\text{Lucy}, \text{Jack}), \text{mother}(\text{Jim}, \text{Betty})) \\ = \{ \text{parent}(\text{Lucy}, \text{Jack}), \text{family}(\text{Lucy}, \text{Jack}) \} \\ \Psi(R_1, \text{father}(\text{Lucy}, \text{Jack}), \text{brother}(\text{Tom}, \text{Joe})) \\ = \{ \text{family}(\text{Lucy}, \text{Jack}) \}$$

となる。

したがって、

$$\text{father}(\text{Lucy}, \text{Jack}) : \\ \text{mother}(\text{Jim}, \text{Betty}) \stackrel{R_1}{\triangleright} \text{brother}(\text{Tom}, \text{Joe})$$

\* 文脈によっては“ $R$ のもとで”を省略することがある。

となる。

原子式間の類似性に関して以下の補題が成立する。

**補題 2.1**  $\alpha, \alpha_1, \alpha_2, \alpha_3$  を任意の原子式とするとき、

$$\alpha : \alpha_1 \stackrel{R}{\sqsubseteq} \alpha_2 \text{ かつ}$$

$$\alpha : \alpha_2 \stackrel{R}{\sqsubseteq} \alpha_3 \text{ ならば } \alpha : \alpha_1 \stackrel{R}{\sqsubseteq} \alpha_3,$$

$$\alpha : \alpha_1 \stackrel{R}{\triangleright} \alpha_2 \text{ かつ}$$

$$\alpha : \alpha_2 \stackrel{R}{\triangleright} \alpha_3 \text{ ならば } \alpha : \alpha_1 \stackrel{R}{\triangleright} \alpha_3,$$

$$\alpha : \alpha_1 \stackrel{R}{\triangleright} \alpha_2 \text{ かつ}$$

$$\alpha : \alpha_2 \stackrel{R}{\sqsubseteq} \alpha_3 \text{ ならば } \alpha : \alpha_1 \stackrel{R}{\triangleright} \alpha_3. \quad \square$$

[証明] 定義 2.3 より原子式間の類似性は集合の包含関係として定義されるため明らか。  $\square$

**2.2 原子式集合間の類似性**

原子式集合間の類似性は、その要素である各原子式の類似性から決定する。その際、集合間でどの原子式とどの原子式が対応づけられるかを考慮する必要がある。そこで、原子式集合間の対応を導入する。

**定義 2.4 (原子式集合間の対応)**  $A, B$  を要素数が等しい原子式の集合とする\*とき、 $A$  と  $B$  との対応  $\varphi$  は以下の条件を満たす  $A \times B$  の部分集合である。

- (1) ある代入  $\theta$  が存在し、 $\varphi$  の任意の要素  $(\alpha, \beta)$  に対して、 $\alpha\theta$  と  $\beta\theta$  の引数の個数および値が一致。
- (2)  $A$  の任意の要素  $\alpha$  に対して、 $(\alpha, \beta) \in \varphi$  なる  $\beta \in B$  がただ一つ存在する。同様に、 $B$  の任意の要素  $\beta$  に対して、 $(\alpha, \beta) \in \varphi$  なる  $\alpha \in A$  がただ一つ存在する。  $\square$

例えば  $A_1, B_1$  を各々次に示す原子式の集合とする。

$$A_1 = \{\text{father}(x, y), \text{kills}(y, z)\}$$

$$B_1 = \{\text{mother}(\text{Jim}, \text{Betty}), \text{hurts}(\text{Betty}, \text{Jim})\}$$

このとき、 $A_1$  と  $B_1$  との対応として、2種類の対応  $\varphi_1, \varphi_2$  を考えることができる。

$$\varphi_1 = \{(\text{father}(x, y), \text{mother}(\text{Jim}, \text{Betty})), (\text{kills}(y, z), \text{hurts}(\text{Betty}, \text{Jim}))\}$$

$$\varphi_2 = \{(\text{father}(x, y), \text{hurts}(\text{Betty}, \text{Jim})), (\text{kills}(y, z), \text{mother}(\text{Jim}, \text{Betty}))\}$$

**定義 2.5 (対応の優先)**  $A, B$  を原子式集合とし、 $A$  と  $B$  との二つの対応を  $\varphi_1, \varphi_2$  とする。ここで、 $A$  の要素  $\alpha$  に対して、 $\beta_1, \beta_2$  を  $(\alpha, \beta_1) \in \varphi_1, (\alpha, \beta_2) \in \varphi_2$  により定めると、

- (1)  $A$  の任意の要素  $\alpha$  に対して、

$$\alpha : \beta_1 \stackrel{R}{\sqsubseteq} \beta_2,$$

- (2)  $A$  のある要素  $\alpha$  に対して、

$$\alpha : \beta_1 \stackrel{R}{\triangleright} \beta_2$$

がともに成立するとき、 $\varphi_1$  は  $\varphi_2$  より  $R$  のもとで優先するといひ、

$$\varphi_1 \stackrel{R}{\gg} \varphi_2$$

と記す。  $\square$

対応の優先関係に関しては、以下の定理が成立する。  
**定理 2.1** 二つの原子式集合のすべての対応からなる集合上で、対応の優先関係  $\gg$  は、非反射律、推移律を満たす。  $\square$

[証明] (非反射律)  $A, B$  を原子式集合とし、両者の任意の対応を  $\varphi$  とする。定義 2.4 より  $A$  の任意の要素  $\alpha$  に対して、 $(\alpha, \beta) \in \varphi$  なる  $\beta \in B$  は一意に決定される。一方、定義 2.3 より、 $\alpha : \beta \triangleright \beta$  は成立しない。したがって、 $\varphi \gg \varphi$  は、定義 2.5 の (2) を満たす  $\alpha$  が存在しないため成立せず、非反射律を満たす。

(推移律)  $\varphi_1, \varphi_2, \varphi_3$  を原子式集合  $A, B$  の任意の対応とする。 $\varphi_1 \gg \varphi_2$  とすると、 $(\alpha, \beta_1) \in \varphi_1, (\alpha, \beta_2) \in \varphi_2$  なる  $\beta_1, \beta_2 \in B$  は、定義 2.5 の (1) より  $A$  の任意の要素  $\alpha$  に対して、 $\alpha : \beta_1 \sqsubseteq \beta_2$  となり、定義 2.5 の (2) より  $A$  のある要素  $\alpha$  に対して、 $\alpha : \beta_1 \triangleright \beta_2$  が成立する。 $\varphi_2 \gg \varphi_3$  とすると、 $(\alpha, \beta_2) \in \varphi_2, (\alpha, \beta_3) \in \varphi_3$  なる  $\beta_2, \beta_3$  についても同様に、 $A$  の任意の要素  $\alpha$  に対して、 $\alpha : \beta_2 \sqsubseteq \beta_3$ 、 $A$  のある要素  $\alpha$  に対して、 $\alpha : \beta_2 \triangleright \beta_3$  がそれぞれ成立する。このため補題 2.1 より、任意の  $\alpha$  に対して、 $\alpha : \beta_1 \sqsubseteq \beta_3$  が成立し、また、 $\alpha : \beta_1 \triangleright \beta_3$  を満たす  $\alpha$  が存在する。よって、 $\varphi_1 \gg \varphi_2, \varphi_2 \gg \varphi_3$  ならば  $\varphi_1 \gg \varphi_3$  となり推移律を満足する。  $\square$

非反射律、推移律を満足する関係は、半順序関係から恒等関係を除外したものに相当し、半順序と同様の性質を持つ。

この定理より、次の極大優先対応が定義できる。

**定義 2.6 (極大優先対応)** 原子式集合  $A$  と  $B$  とのある対応  $\varphi$  に対して、 $\varphi$  より優先する他の  $A$  と  $B$  との対応が存在しないとき、 $\varphi$  を  $A$  と  $B$  との  $R$  における極大優先対応という。  $\square$

既に述べたように二つの原子式集合間には複数の対応を考えることができるが、それらの中で、より類似した原子式の組み合わせをもつ対応が優先した対応となる。

したがって、極大優先対応は、二つの原子式集合各々に存在する原子式を同一の変数に異なる値が束縛されることがない範囲で最も類似するように組み合わせた

\* 以下、二つの原子式集合間の対応に関して論じる際には、両者の要素数の一致を前提とする。

対応を意味する。

上に示した例では,  $\text{father}(x,y)$  は  $R_1$  のもとで  $\text{hurts}(\text{Betty}, \text{Jim})$  より  $\text{mother}(\text{Jim}, \text{Betty})$  に類似しており, 同様に  $\text{kills}(y,z)$  は  $R_1$  のもとで  $\text{mother}(\text{Jim}, \text{Betty})$  より  $\text{hurts}(\text{Betty}, \text{Jim})$  に類似しているため, 対応  $\varphi_1$  は  $\varphi_2$  よりも優先する. したがって  $R_1$  に関する  $A_1$  と  $B_1$  との極大優先対応は  $\varphi_1$  となる.

**定義 2.7 (原子式集合間の類似性)**  $A, B, C$  をそれぞれ原子式の集合とし,  $A$  と  $B$  との任意の極大優先対応, および  $A$  と  $C$  との任意の極大優先対応をそれぞれ  $\varphi_{AB}, \varphi_{AC}$  とする. ここで,  $A$  の要素  $\alpha$  に対して,  $\beta_B, \beta_C$  を  $(\alpha, \beta_B) \in \varphi_{AB}, (\alpha, \beta_C) \in \varphi_{AC}$  により定めると,

(1)  $A$  の任意の要素  $\alpha$  に対して,

$$\alpha : \beta_B \stackrel{R}{\supseteq} \beta_C,$$

(2)  $A$  のある要素  $\alpha$  に対して,

$$\alpha : \beta_B \stackrel{R}{\triangleright} \beta_C$$

が成立するとき, 原子式集合  $A$  は  $R$  のもとで  $C$  よりも  $B$  に類似しているといい,

$$A : B \stackrel{R}{\triangleright} C$$

と記す. また, 上式が成立しないとき,

$$A : B \not\stackrel{R}{\triangleright} C$$

と記す.

いま, 次に示す原子式集合  $C_1$  を考える.

$$C_1 = \{\text{brother}(\text{Tom}, \text{Joe}), \\ \text{strikes}(\text{Joe}, \text{Mark})\}$$

$R_1$  に関する  $A_1$  と  $C_1$  との極大優先対応は

$$\{(\text{father}(x,y), \text{brother}(\text{Tom}, \text{Joe})), \\ (\text{kills}(y,z), \text{strikes}(\text{Joe}, \text{Mark}))\},$$

となる. したがって,

$$A_1 : B_1 \stackrel{R_1}{\triangleright} C_1$$

が成立する.

原子式集合間の類似性に関して, 次の定理が成立する.

**定理 2.2** 原子式集合の集合上で, ある原子式集合に対する原子式集合間の類似性は, 非反射律, 推移律を満たす. □

[証明] (非反射律)  $A$  をある原子式集合,  $B$  を任意の原子式集合とする.  $A$  と  $B$  の任意の極大優先対応を  $\varphi$  としたとき, 定義 2.4 より  $A$  の任意の要素

$\alpha$  に対して,  $(\alpha, \beta) \in \varphi$  なる  $\beta$  は一意に決定される. 一方, 定義 2.3 より,  $\alpha : \beta \triangleright \beta$  は成立しない. したがって,  $A : B \stackrel{R}{\triangleright} B$  は, 定義 2.7 の (2) を満たす  $\alpha$  が存在しないため成立せず, 非反射律を満たす.

(推移律)  $A$  をある原子式集合,  $B, C, D$  を任意の原子式集合とし,  $A$  と  $B, A$  と  $C, A$  と  $D$  の任意の極大優先対応をそれぞれ  $\varphi_{AB}, \varphi_{AC}, \varphi_{AD}$  とする.  $A : B \stackrel{R}{\triangleright} C$  とすると,  $(\alpha, \beta_B) \in \varphi_{AB}, (\alpha, \beta_C) \in \varphi_{AC}$  なる  $\beta_B, \beta_C$  は, 定義 2.7 の (1) より  $A$  の任意の要素  $\alpha$  に対して,  $\alpha : \beta_B \supseteq \beta_C$  となり, 定義 2.7 の (2) より  $A$  のある要素  $\alpha$  に対して,  $\alpha : \beta_B \triangleright \beta_C$  が成立する.  $A : C \stackrel{R}{\triangleright} D$  とすると,  $(\alpha, \beta_C) \in \varphi_{AC}, (\alpha, \beta_D) \in \varphi_{AD}$  なる  $\beta_C, \beta_D$  についても同様に,  $A$  の任意の要素  $\alpha$  に対して,  $\alpha : \beta_C \supseteq \beta_D, A$  のある要素  $\alpha$  に対して,  $\alpha : \beta_C \triangleright \beta_D$  がそれぞれ成立する. このため補題 2.1 より, 任意の  $\alpha$  に対して,  $\alpha : \beta_B \supseteq \beta_D$  が成立し, また,  $\alpha : \beta_B \triangleright \beta_D$  を満たす  $\alpha$  が存在する. よって, 推移律を満足する. □

この定理は, いくつかの原子式集合が与えられたときに, その中の任意の原子式集合に対して, 極大に類似した原子式集合が決定可能であることを示している.

### 2.3 類推的一般化の定式化

ここでは類推的一般化を定式化する. 学習の目標概念を表現した変数を含む原子式を  $\delta$ , 目標概念に関連した変数を含まない事実の集合からなる例題を  $E$  とする. ただし, 例題  $E$  は  $\delta$  と同じ述語の基礎原子式 (目標例と呼ぶ)  $\delta'$  を含むものとする. また, 既存のルール集合を  $R$  とする. 例題  $E$  から目標例  $\delta'$  を除いた集合を  $E'$  とし,  $R \cup E$  から演繹可能な変数を含まない原子式の集合を  $E''$  とする. このとき類推的一般化は次のように定義される.

**定義 2.8 (類推的一般化)**  $R, E, \delta$  に対して,

$$R \cup E' \not\vdash \delta' \tag{1}$$

が成立するとき,

$$R \cup E' \cup \{r\} \vdash \delta' \tag{2}$$

を満たし,

$$\text{cons}(r)\theta = \delta' \tag{3}$$

が成立する代入  $\theta$  に対して, 以下の条件を満足するベースルール  $r' (\in R)$  が存在するルール  $r$  を生成することを類推的一般化という. ただし,  $\Pi(r)$  は  $r$  を構成するすべての原子式を要素とする集合を表す.

1.  $(\text{cons}(r'), \delta')$  を含む  $\Pi(r')$  から  $\Pi(r)\theta$  への対応  $\varphi$  が存在し,

$$\varphi'\theta = \varphi\theta$$

なる  $\Pi(r')$  から  $\Pi(r)$  への対応を  $\varphi'$  とすると,

$$\forall(\alpha, \beta) \in \varphi'$$

に対して、 $\alpha$  と  $\beta$  の引数の個数、値は一致する。

2. 任意の原子式集合  $A (\subseteq E'')$  に対して、次式が成立する。

$$\Pi(r') : A \not\stackrel{R}{\sim} \Pi(r)\theta$$

3. 任意のルール  $r'' (\in R)$  に対して、次式が成立する。

$$\Pi(r)\theta : \Pi(r'') \not\stackrel{R}{\sim} \Pi(r')$$

□

式 (1), (2) は仮説推論の一般的な枠組を用いて、帰納的学習問題を定式化したものである。なお、ここでは取り扱う論理式をホーン節集合のみに限定しているため、論理式集合の無矛盾性は維持されていることに注意されたい。

一般にこれら式 (1), (2) を満足するルールは多数存在する。そこで、類推的一般化では、前述の原子式集合間の類似性に基づき、例題を構成する原子式集合と既存のルールを構成する原子式集合との類似性を評価し、その類似性の極大化を図る。すなわち、条件 3 により最も一般化の際のガイドとなるベースルールを発見すると同時に、条件 2 により、ベースルールを構成する原子式集合に類似した例題の部分集合が抽出される。条件 1 は、その前提となる原子式集合間の対応の存在を意図したものである。

定理 2.2 により、既存のルールを構成する原子式集合と例題の部分集合との間に、ルールの帰結部と目標例との対を前提とした対応が存在する限り、一つ以上のベースルール  $r'$  ならびに  $\Pi(r)$  の存在が保証される。そこで、得られたベースルールに基づき、抽出された原子式における定数を変数化することにより、式 (1), (2) を満たす新たなルールが生成される。なお、式 (3) は、生成すべきルールの結論部と目標例が同一の述語記号を用いて表現されるよう制限するものである。

#### 2.4 類推的一般化によるルール生成例

類推的一般化によるルール生成を簡単な例を使って説明する。

以下に示す  $R_2$  は家族関係を定義した七つのルールからなる集合、 $E_1$  は学習を図る目標概念 “grandmother( $s, t$ )” の例題である。

$$R_2 = \{ \text{grandfather}(x, z) \leftarrow \text{parent}(x, y), \dots (r1) \\ \text{father}(y, z), \\ \text{uncle}(x, z) \leftarrow \text{parent}(x, y), \text{brother}(y, z), \dots (r2) \\ \text{cousin}(x, y) \leftarrow \text{parent}(x, v), \text{parent}(y, w), \dots (r3) \\ \text{brother}(v, w), \dots (r3) \}$$

$$\text{parent}(x, y) \leftarrow \text{mother}(x, y), \dots (r4) \\ \text{parent}(x, y) \leftarrow \text{father}(x, y), \dots (r5) \\ \text{family}(x, y) \leftarrow \text{parent}(x, y), \dots (r6) \\ \text{family}(x, y) \leftarrow \text{brother}(x, y) \dots (r7)$$

$$E_1 = \{ \text{grandmother}(\text{Peter}, \text{Mary}), \text{mother}(\text{Paul}, \text{Mary}), \\ \text{father}(\text{Peter}, \text{Paul}), \text{mother}(\text{Peter}, \text{Lucy}), \\ \text{likes}(\text{Paul}, \text{Mary}), \text{engineer}(\text{Peter}), \\ \text{student}(\text{Paul}) \}$$

既存ルール集合  $R_2$ , 例題  $E_1$  に対し、 $R_2 \cup E_1$  から演繹可能な原子式の集合は以下の  $E_1''$  となる。

$$E_1'' = \{ \text{grandmother}(\text{Peter}, \text{Mary}), \text{mother}(\text{Paul}, \text{Mary}), \\ \text{parent}(\text{Paul}, \text{Mary}), \text{family}(\text{Paul}, \text{Mary}), \\ \text{father}(\text{Peter}, \text{Paul}), \text{parent}(\text{Peter}, \text{Paul}), \\ \text{family}(\text{Peter}, \text{Paul}), \text{mother}(\text{Peter}, \text{Lucy}), \\ \text{parent}(\text{Peter}, \text{Lucy}), \text{family}(\text{Peter}, \text{Lucy}), \\ \text{likes}(\text{Paul}, \text{Mary}), \text{engineer}(\text{Peter}), \\ \text{student}(\text{Paul}) \}$$

類推的一般化の定義における式 (1)~(3), および条件 1 から  $E_1''$  の部分集合と  $R_2$  内のルールを構成する原子式集合との間に対応が存在するもののみが、考慮対象となることがわかる。例えば、ルール  $r1$  を構成する原子式との間に対応が存在する  $E_1''$  の部分集合は全部で 12 通りある。以下にその一部を示す。

$$\{ \text{grandmother}(\text{Peter}, \text{Mary}), \text{father}(\text{Peter}, \text{Paul}), \\ \text{mother}(\text{Paul}, \text{Mary}) \} \dots (s1) \\ \{ \text{grandmother}(\text{Peter}, \text{Mary}), \text{father}(\text{Peter}, \text{Paul}), \\ \text{parent}(\text{Paul}, \text{Mary}) \} \dots (s2) \\ \{ \text{grandmother}(\text{Peter}, \text{Mary}), \text{father}(\text{Peter}, \text{Paul}), \\ \text{family}(\text{Paul}, \text{Mary}) \} \dots (s3) \\ \{ \text{grandmother}(\text{Peter}, \text{Mary}), \text{parent}(\text{Peter}, \text{Paul}), \\ \text{mother}(\text{Paul}, \text{Mary}) \} \dots (s4)$$

これらの集合に関しては、例えば

$$\Pi(r1) : s1 \stackrel{R_2}{\sim} s2$$

という関係が成立するため、類推的一般化の定義の条件 2 より、 $r1$  をベースルールと仮定した場合、部分集合  $s2$  からはルールが生成されないことがわかる。同様の評価に基づき、 $r1$  に対して、部分集合  $s1, s4$  のみがルール生成対象の候補として残る。ルール  $r2$  に対しても同様に、集合  $s1, s4$  が採用される。なお、ルール  $r3 \sim r7$  に対しては対応が存在する集合は得られない。

一方、 $\Pi(r1)$  と  $\Pi(r2)$  の類似性を評価すると、

$$s1 : \Pi(r1) \stackrel{R_2}{\sim} \Pi(r2)$$

$$s4 : \Pi(r1) \stackrel{R_2}{\sim} \Pi(r2)$$

という関係が成立するため、類推的一般化の定義における条件 3 に基づき、ルール  $r2$  はベースルールとなり得ないことがわかる。

この時点で、

$$\{ (\text{grandfather}(x, z), \text{grandmother}(\text{Peter}, \text{Mary})),$$

```
(parent(x, y), father(Peter, Paul)),
(father(y, z), mother(Paul, Mary))}
{(grandfather(x, z), grandmother(Peter, Mary)),
(parent(x, y), parent(Peter, Paul)),
(father(y, z), mother(Paul, Mary))}
```

の二つの対応が存在し、この対応に従った定数変数化により原子式集合  $s1'$ ,  $s4'$  が得られる。

```
{grandmother(x, z), father(x, y), mother(y, z)}
... (s1')
{grandmother(x, z), parent(x, y), mother(y, z)}
... (s4')
これら集合をもとにルール r8, r9 が生成される。
grandmother(x, z) ← father(x, y), mother(y, z)
... (r8)
grandmother(x, z) ← parent(x, y), mother(y, z)
... (r9)
```

ルール r8, r9 はともに、類推的一般化の定義を満たすルールである。両者とも誤ったルールではないと言う意味で目標概念を定義した正しいルールとなっている。しかしながら、r9 がベースルールとして用いられた r1 と相対なルールであるのに対し、r8 は父方の祖母のみを定義している点で不十分であり、r9の方がより妥当なルールと言える。

### 3. 実験と考察

#### 3.1 構文解析ルールの生成実験

類推的一般化による学習の特性を明らかにするため、簡単な英文の構文解析システムにおけるルールの学習実験を行った。対象とする構文解析システムは、表1に示す文法規則に基づいて英文の構文解析を行うものである。このシステムでは、文はリストとして扱われる。例えば、文“*Jack likes Mary*”は次のようにリスト表現される。

```
[jack, likes, mary]
```

また、

```
noun([jack, likes, mary], [likes, mary])
```

は *jack* が名詞 (noun) であることを示している。システムは文法規則を用いて後向きに推論することにより、与えられた文が文法的に正しいかどうかを判定する。

実験に用いた既存のルール集合 RS1~RS4 を表1に示す。

ここでは、既存のルールを変化させながら、様々な構文を目標概念として設定し、その目標概念の例題となる文の一つずつ与えることにより新たなルールの生成を試みた。

表2に実験結果を示す。表には、前提とした既存のルール集合、および与えられた目標概念とその例題文に対して、生成されたルールを示している。

なお、例題文は、例えば、文“*I was rich.*”を  

```
{ past_sentence([i, was, rich], []),
i([i, was, rich], [was, rich]),
was([was, rich], [rich]), rich([rich], [])}
```

のように表現して与え、また、「 $\text{noun}(s, e) \leftarrow i(s, e)$ 」のような単語辞書としてのルールが適宜利用可能と仮定した。

#### 3.2 実験結果に対する考察

表2に示した実験結果からわかるように、類推的一般化により、単一の例題から新たなルールの生成が可能である。

実験1~4は、前提となる既存のルール集合を変えながら、同じ文章を与えて単純な過去形の構文規則の学習を試みたものである。実験1は、基本文型に相当する構文規則のみを有する状況、実験2は、それに加えて、現在形の文章の構文規則を有する状況、実験3は、最も基本的な構文規則と動詞の種類に関する知識を持つ状況をそれぞれ想定している。いずれの実験においても、正しい過去形の構文規則が生成されていない。一方、現在形の構文規則と動詞の種類に関する知識を与えた実験4では、例題文に対応した正しい規則  $r_{41}$  の生成に成功している。

また、実験5~8は、同様に既存のルールを変化させながら、さらに複雑な受動態現在文の構文規則の学習を試みたものである。実験5~7においては、妥当なルールは生成されていない。その理由は、これらの実験では、二つ以上の動詞が用いられることがない簡単な構文に関するルールのみが存在を前提としているためである。ルール集合 RS3 には、より複雑な現在進行形に関するルールが含まれており、これを既存の知識と仮定した実験8では、正しいルール  $r_{81}$  が生成されている。受動態現在文のルールを正しく学習するためには、助動詞と本動詞が結合して一つの動詞としての働きをすることを知っている必要があり、類似した構造を持つ現在進行形の知識を備えることによつて、これが実現されていることがわかる。

これらの実験から、類推的一般化による学習では、ある目標概念のルールを学習する際に、その概念に対比できるような類似したルールが既知であり、かつ、両者の概念間の差異に関する背景知識を有することが必要となることがわかる。すなわち前に示した実験では、目標概念である過去形構文や受動態現在の構文に対比可能な現在形構文や現在進行形構文に関するルールを有し、両構文間の差異の根拠を与える動詞に関する分類ルールが必要であることが相当する。一方、人間が例えば過去形の構文を学習する場合においても、

表 1 既存の構文ルール  
Table 1 Existing rules.

RS1	$\text{sentence}(s, e) \leftarrow \text{noun}(s, v_1), \text{verb}(v_1, e).$ $\text{sentence}(s, e) \leftarrow \text{noun}(s, v_1), \text{verb}(v_1, v_2), \text{adjective}(v_2, e).$ $\text{sentence}(s, e) \leftarrow \text{noun}(s, v_1), \text{verb}(v_1, v_2), \text{noun}(v_2, e).$ $\text{sentence}(s, e) \leftarrow \text{noun}(s, v_1), \text{verb}(v_1, v_2), \text{noun}(v_2, v_3), \text{noun}(v_3, e).$ $\text{sentence}(s, e) \leftarrow \text{noun}(s, v_1), \text{verb}(v_1, v_2), \text{noun}(v_2, v_3), \text{adjective}(v_3, e).$
RS2	$\text{present\_sentence}(s, e) \leftarrow \text{noun}(s, v_1), \text{present\_main\_verb}(v_1, e).$ $\text{present\_sentence}(s, e) \leftarrow \text{noun}(s, v_1), \text{present\_main\_verb}(v_1, v_2), \text{adjective}(v_2, e).$ $\text{present\_sentence}(s, e) \leftarrow \text{noun}(s, v_1), \text{present\_main\_verb}(v_1, v_2), \text{noun}(v_2, e).$ $\text{present\_sentence}(s, e) \leftarrow \text{noun}(s, v_1), \text{present\_main\_verb}(v_1, v_2), \text{noun}(v_2, v_3), \text{noun}(v_3, e).$ $\text{present\_sentence}(s, e) \leftarrow \text{noun}(s, v_1), \text{present\_main\_verb}(v_1, v_2), \text{noun}(v_2, v_3), \text{adjective}(v_3, e).$
RS3	$\text{present\_progressive\_sentence}(s, e)$ $\quad \leftarrow \text{noun}(s, v_1), \text{present\_be}(v_1, v_2), \text{present\_participle}(v_2, e).$ $\text{present\_progressive\_sentence}(s, e)$ $\quad \leftarrow \text{noun}(s, v_1), \text{present\_be}(v_1, v_2), \text{present\_participle}(v_2, v_3), \text{adjective}(v_3, e).$ $\text{present\_progressive\_sentence}(s, e)$ $\quad \leftarrow \text{noun}(s, v_1), \text{present\_be}(v_1, v_2), \text{present\_participle}(v_2, v_3), \text{noun}(v_3, e).$ $\text{present\_progressive\_sentence}(s, e)$ $\quad \leftarrow \text{noun}(s, v_1), \text{present\_be}(v_1, v_2), \text{present\_participle}(v_2, v_3), \text{noun}(v_3, v_4), \text{noun}(v_4, e).$ $\text{present\_progressive\_sentence}(s, e)$ $\quad \leftarrow \text{noun}(s, v_1), \text{present\_be}(v_1, v_2), \text{present\_participle}(v_2, v_3), \text{noun}(v_3, v_4), \text{adjective}(v_4, e).$
RS4	$\text{verb}(s, e) \leftarrow \text{main\_verb}(s, e).$ $\text{verb}(s, e) \leftarrow \text{auxiliary\_verb}(s, e).$ $\text{verb}(s, e) \leftarrow \text{participle}(s, e).$ $\text{main\_verb}(s, e) \leftarrow \text{present\_main\_verb}(s, e).$ $\text{main\_verb}(s, e) \leftarrow \text{past\_main\_verb}(s, e).$ $\text{auxiliary\_verb}(s, e) \leftarrow \text{present\_auxiliary\_verb}(s, e).$ $\text{auxiliary\_verb}(s, e) \leftarrow \text{past\_auxiliary\_verb}(s, e).$ $\text{participle}(s, e) \leftarrow \text{present\_participle}(s, e).$ $\text{participle}(s, e) \leftarrow \text{past\_participle}(s, e).$ $\text{present\_main\_verb}(s, e) \leftarrow \text{present\_be}(s, e).$ $\text{present\_auxiliary\_verb}(s, e) \leftarrow \text{present\_be}(s, e).$ $\text{past\_main\_verb}(s, e) \leftarrow \text{past\_be}(s, e).$ $\text{past\_auxiliary\_verb}(s, e) \leftarrow \text{past\_be}(s, e).$ $\text{present\_main\_verb}(s, e) \leftarrow \text{present\_have}(s, e).$ $\text{present\_auxiliary\_verb}(s, e) \leftarrow \text{present\_have}(s, e).$ $\text{past\_main\_verb}(s, e) \leftarrow \text{past\_have}(s, e).$ $\text{past\_auxiliary\_verb}(s, e) \leftarrow \text{past\_have}(s, e).$

現在形の構文に関する知識のみが既知である場合や、現在形の構文に関しては未知で、基本文型と動詞の時制に関する知識のみを持つ場合には、過去形の構文について理解することは困難である。

以上のことから、類推的一般化は、完全に未知の概念に関するルール、あるいは既存ルールと大きく構造が異なるルールの学習は困難であるが、関連概念や類似知識を備えることにより、単一の例題からでも妥当なルールを導くことが可能であり、その学習過程は、ある程度人間の学習過程に近いといえる。

なお、すべての実験に対して、 $r_{12}, r_{13}, r_{22}$  などの不適切なルールが生成されている。これは、現在の類推的一般化の枠組で導入した類似性が線形順序でないため、極度に類似した複数の既存ルールをベースルールとして採用することに起因する。これに対しては、現

在のところ、ルールを生成後、人間がインタラクティブにルールを吟味する方策を想定しており、この場合、問題対象に応じた対処が必要となる。例えば、今回の実験においては、「 $\text{verb}(s, e) \leftarrow \text{main\_verb}(s, e)$ 」のように前提部が少数の原子式から構成されるルールをベースルールとして用いたときに、無意味なルールが生成される傾向があることから、そのようなベースルールの選択順位を下げたり、あるいは、「 $\text{sentence}(s, v_1)$ 」という記述を含むなど、生成されたルールが構文的に明らかな誤りをもつ場合には排除するなどの処理により、有用なルールのみでの選別が可能となる。

不適切なベースルールの選択を根本的に抑制する一方策として、原子式集合を構成している各原子式の類似性に応じた重みを求め、これらを合計した値を用いる等、原子式集合全体としての類似性の評価に定量的



表 2 生成された構文ルール

Table 2 Generated rules.

実験 1	既存ルール集合 目標概念 例題文 生成ルール	RS1 $\text{past\_sentence}(s, e)$ (過去文) "I was rich." $r_{11}: \text{past\_sentence}(s, e) \leftarrow \text{noun}(s, v_1), \text{verb}(v_1, v_2), \text{adjective}(v_2, e).$ $r_{12}: \text{past\_sentence}(s, e) \leftarrow \text{sentence}(s, v_1), \text{adjective}(v_1, e).$ $r_{13}: \text{past\_sentence}(s, e) \leftarrow \text{sentence}(s, e).$
実験 2	既存ルール集合 目標概念 例題文 生成ルール	RS1 $\cup$ RS2 $\text{past\_sentence}(s, e)$ (過去文) "I was rich." $r_{21}: \text{past\_sentence}(s, e) \leftarrow \text{noun}(s, v_1), \text{verb}(v_1, v_2), \text{adjective}(v_2, e).$ $r_{22}: \text{past\_sentence}(s, e) \leftarrow \text{sentence}(s, v_1), \text{adjective}(v_1, e).$ $r_{23}: \text{past\_sentence}(s, e) \leftarrow \text{sentence}(s, e).$
実験 3	既存ルール集合 目標概念 例題文 生成ルール	RS1 $\cup$ RS4 $\text{past\_sentence}(s, e)$ (過去文) "I was rich." $r_{31}: \text{past\_sentence}(s, e) \leftarrow \text{noun}(s, v_1), \text{verb}(v_1, v_2), \text{adjective}(v_2, e).$ $r_{32}: \text{past\_sentence}(s, e) \leftarrow \text{sentence}(s, v_1), \text{adjective}(v_1, e).$ $r_{33}: \text{past\_sentence}(s, e) \leftarrow \text{sentence}(s, e).$
実験 4	既存ルール集合 目標概念 例題文 生成ルール	RS1 $\cup$ RS2 $\cup$ RS4 $\text{past\_sentence}(s, e)$ (過去文) "I was rich." $r_{41}: \text{past\_sentence}(s, e) \leftarrow \text{noun}(s, v_1), \text{past\_main\_verb}(v_1, v_2), \text{adjective}(v_2, e).$ $r_{42}: \text{past\_sentence}(s, e) \leftarrow \text{sentence}(s, v_1), \text{adjective}(v_1, e).$ $r_{43}: \text{past\_sentence}(s, e) \leftarrow \text{sentence}(s, e).$
実験 5	既存ルール集合 目標概念 例題文 生成ルール	RS1 $\text{present\_passive\_voice\_sentence}(s, e)$ (受動態現在文) "I am called Jack." $r_{51}: \text{present\_passive\_voice\_sentence}(s, e)$ $\leftarrow \text{noun}(s, v_1), \text{verb}(v_1, v_2), \text{verb}(v_2, v_3), \text{noun}(v_3, e).$ $r_{52}: \text{present\_passive\_voice\_sentence}(s, e)$ $\leftarrow \text{sentence}(s, v_1), \text{verb}(v_1, v_2), \text{noun}(v_2, e).$
実験 6	既存ルール集合 目標概念 例題文 生成ルール	RS1 $\cup$ RS2 $\text{present\_passive\_voice\_sentence}(s, e)$ (受動態現在文) "I am called Jack." $r_{61}: \text{present\_passive\_voice\_sentence}(s, e)$ $\leftarrow \text{noun}(s, v_1), \text{present\_main\_verb}(v_1, v_2), \text{verb}(v_2, v_3), \text{noun}(v_3, e).$ $r_{62}: \text{present\_passive\_voice\_sentence}(s, e)$ $\leftarrow \text{present\_sentence}(s, v_1), \text{verb}(v_1, v_2), \text{noun}(v_2, e).$
実験 7	既存ルール集合 目標概念 例題文 生成ルール	RS1 $\cup$ RS2 $\cup$ RS4 $\text{present\_passive\_voice\_sentence}(s, e)$ (受動態現在文) "I am called Jack." $r_{71}: \text{present\_passive\_voice\_sentence}(s, e)$ $\leftarrow \text{noun}(s, v_1), \text{present\_main\_verb}(v_1, v_2), \text{verb}(v_2, v_3), \text{noun}(v_3, e).$ $r_{72}: \text{present\_passive\_voice\_sentence}(s, e)$ $\leftarrow \text{present\_sentence}(s, v_1), \text{verb}(v_1, v_2), \text{noun}(v_2, e).$
実験 8	既存ルール集合 目標概念 例題文 生成ルール	RS1 $\cup$ RS2 $\cup$ RS3 $\cup$ RS4 $\text{present\_passive\_voice\_sentence}(s, e)$ (受動態現在文) "I am called Jack." $r_{81}: \text{present\_passive\_voice\_sentence}(s, e)$ $\leftarrow \text{noun}(s, v_1), \text{present\_be}(v_1, v_2), \text{past\_participle}(v_2, v_3), \text{noun}(v_3, e).$ $r_{82}: \text{present\_passive\_voice\_sentence}(s, e)$ $\leftarrow \text{present\_sentence}(s, v_1), \text{verb}(v_1, v_2), \text{noun}(v_2, e).$

かつ絶対的尺度の導入がある。最も端的な定量的類似性評価尺度の導入形態として、ベースルールと対応付けられる各原子式について、述語記号が一致していれば類似していると見なし、重み 1 を与え、そうでな

い場合は重み 0 とし、それを合計し、値が最大となる対応のみを採用することが考えられる。このような、簡単な評価尺度であっても、2.4 節に示した目標概念 "grandmother(s, t)" のルール学習においては、妥当

な  $r_9$  のみを選択できる。また、表 2 に示した英文構文解析ルールのうち、 $r_{12}$ ,  $r_{13}$ ,  $r_{22}$ ,  $r_{23}$ ,  $r_{32}$ ,  $r_{33}$ ,  $r_{42}$ ,  $r_{43}$ ,  $r_{52}$ ,  $r_{62}$ ,  $r_{72}$ ,  $r_{82}$  は同じ目標概念に対して生成された他のルールに比べて小さな重みをとるため除外され、結果として適切なルールのみ生成が可能である。

### 3.3 関連手法との比較

類推の一般化は、例題として与えられた事例に類似した既存の知識を利用するという点において、類推や事例ベース推論 (CBR: Case Based Reasoning) と関連が深いと考えられる。そこで、ここでは類推に関する研究との比較、CBR における類似性評価との比較を通して、類推的一般化の位置付けを明確にする。

#### 3.3.1 類推との比較

類推に関するこれまでの定式化のほとんどが、ソース (既知の事実からなる世界) とターゲット (新たに与えられた事実からなる世界) との類似性をもとにターゲットにおける未知の事実を推測する<sup>7)</sup>、あるいはソースにおいて成立する知識の中からターゲットにおいても成り立つものを発見し、問題解決に利用する<sup>8)</sup>ための枠組となっている。帰納的学習に類推を利用した例として、De Raedt らの類推による構成的帰納 (constructive induction by analogy)<sup>9)</sup>を挙げることができるが、この手法はルールの構造を 2 階述語形式のスキーマで捉え、これを用いて、類似した構造を有するルールを生成する際に類推を用いたものであり、一般化の際には、通常の帰納的学習と同様に多数の例題を必要とする。このように、帰納的学習における主要な操作である例題の一般化の際に類推を利用した方式は筆者の知る範囲では見あたらず、この意味から類推的一般化は新しい枠組と思われる。

また、類推を実現する上で重要な役割を果たす類似性評価において、既存手法のほとんどが、述語記号の一致<sup>10),11)</sup>や、ソースとターゲットとの間の明示的な対応付け<sup>8)</sup>などを採用しているのに対し、類推的一般化では、演繹的閉包の包含関係を利用することにより、述語記号の一致がみられないような既存の類似知識を用いたルールの生成が可能である。

述語記号の一致に基づく類似性で、これと同様の処理を実現する方策として、Tenenbergs の抽象化<sup>12)</sup>において見られる述語写像を利用し、異なる述語を同一化する手法が考えられる。そのような手法と比較した提案手法の特徴として、類似性の相対的な順序付けが可能となることが挙げられる。すなわち、類推的一般化は、常にその時点において最も類似したベースルールをもとにルールを生成する。したがって、より適切なルールが既存のルール集合内に追加された場合には、その

ようなルールをベースルールとすることが可能となる。逆に言えば、必ずしも最適なベースルールが存在しない場合にも、その次に例題に類似しているルールを用いることにより、何らかのルールを生成することができる。このような類似性に基づく順序づけは、述語記号の同一性などによる類似性の評価では困難であり、類推的一般化の大きな特徴の一つである。

#### 3.3.2 CBR との比較

CBR は、与えられた問題に類似する既知の事例を直接利用することにより、その問題の解決を実現する枠組である。このため、帰納的に一般的な規則を学習する類推的一般化とは既知知識の利用目的は異なるが、何らかの類似性評価尺度に基づいて多数の既知知識の中から利用可能なものを検索するという点では共通している。

CBR における代表的な類似性評価手法として、問題や事例をいくつかの属性で表現し、あらかじめ領域知識として定義した概念階層関係におけるパス上の距離で評価する手法<sup>13)</sup>、現在の問題と事例との間における属性やその値の一致の度合により評価する手法<sup>14),15)</sup>などが挙げられる。

概念階層関係におけるパスを用いる手法では、あらかじめ明示的に類似性の基準となる領域依存の知識を対象ごとに構築する必要がある。類推的一般化における既存知識にもこういった性格の知識が内在している必要はあるが、とくに意図的に類似性を判断するための知識を区別して与えることなく、対象の性質を列挙することにより、類似性評価が可能である。

属性やその値の一致による類似性評価においても、どの属性を重視するかなど、領域固有の情報が重要な役割を果たすのに対し、類推的一般化は、既存知識の演繹的閉包に基づく類似性評価により、領域に依存しない一般的な枠組を与えることを狙いとしている。もちろん、3.2 節で述べたように、類推的一般化においても、より良い学習結果を得るためには、類似性に関する絶対的、あるいは定量的な尺度を併用する余地が残されており、実問題に適用する際には、対象に応じて CBR 研究で培われた各種技術の導入も必要となろう。

## 4. む す び

類推を応用した帰納的なルール獲得の枠組として類推的一般化を定式化した。また、学習例を通してその特徴を明らかにした。提案手法は

1. 既存のルールベース内のルールをガイドとして用いることにより単一の例題からルールを獲得することが可能である。

2. 例題とルール間の類似性が、各々を構成する原子式の演繹的閉包間の包含関係に基づいて定式化されており、類似ルールを形式的な操作により選択することが可能である。
3. 上記の尺度を用いることにより、相対的な順序づけに基づいた類似性の評価が可能である。
4. 類似性を明示的に与えることなく類似性の評価が可能である。

などの特徴を持つ。今後の課題として、

1. 類推的一般化を基本原理とする効率的な学習手続きの開発、およびその計算論的評価、
2. 類似性評価に対する定量的尺度の導入による不適切なルール生成の抑制、
3. 複数のルールをベースルールとして用いた、複数ルールの同時学習が可能な学習の枠組の設定

などが残されている。

謝辞 日頃御指導を賜る大阪大学教授・北橋忠宏先生、ならびに同学教授・薦田憲久先生に深謝する。また、実験に御助力いただいた本学卒業生の森 敏昭氏、保坂大輔氏に感謝する。なお、本研究の一部は文部省科学研究費の補助によるものである。

### 参考文献

- 1) Dietterich, T.G. and Michalski, R.S.: A Comparative Review of Selected Methods for Learning from Examples, *Machine Learning*, Michalski, R.S., Carbonell, J.G. and Mitchell, T.M. (Eds.), pp.41-81, Morgan Kaufmann (1983).
- 2) Michalski, R.S.: A Theory and Methodology of Inductive Learning, *Machine Learning*, Michalski, R.S., Carbonell, J.G. and Mitchell, T.M. (Eds.), pp.83-134, Morgan Kaufmann (1983).
- 3) Mitchell, T.M., Keller, R.M. and Kedar-Cabelli, S.T.: Explanation-based Generalization: A Unifying View, *Machine Learning*, Vol.1, No.1, pp.47-80 (1986).
- 4) Russell, S.J.: Analogy and Single-instance Generalization, *Proc. 4th Int'l Workshop on Machine Learning*, pp.390-397 (1987).
- 5) Ohkawa, T., Mori, T., Babaguchi, N. and Tezuka, Y.: Analogical Generalization, *Proc. Int'l Conf. on Fifth Generation Computer Systems 1992*, pp.497-504 (1992).
- 6) Muggleton, S. and Feng, C.: Efficient Induction of Logic Programs, *Proc. 1st Int'l Workshop on Algorithmic Learning Theory*, pp.368-381 (1990).
- 7) Winston, P.H.: Learning and Reasoning by Analogy, *Comm. ACM*, Vol.23, No.12, pp.689-703 (1980).
- 8) Greiner, R.: Learning by Understanding Analogies, *Artif. Intell.*, Vol.35, pp.81-125 (1988).
- 9) De Raedt, L. and Bruynooghe, M.: Constructive Induction by Analogy, *Proc. 6th Int'l Workshop on Machine Learning*, pp.476-477 (1989).
- 10) Haraguchi, M.: Towards a Mathematical Theory of Analogy, *Bull. of Inform. Cybernetics*, Vol.21 (1985).
- 11) Arima, J.: Logical Structure of Analogy: Preliminary Report, *Proc. Int'l Conf. on Fifth Generation Computer Systems 1992*, pp.505-513 (1992).
- 12) Tenenber, J.D.: Preserving Consistency across Abstraction Mappings, *Proc. 10th Int'l Joint Conf. on Artificial Intelligence*, pp.1011-1014 (1987).
- 13) 新田克己: 法的推論システム HELIC-II, 人工知能学会誌, Vol.7, No.4, pp.603-607 (1992).
- 14) 中村孝太郎, 小林重信: 事例ベース推論の対話型モデルとその機械調整支援への適用, 人工知能学会誌, Vol.4, No.6, pp.704-713 (1989).
- 15) 奥田健三, 山崎勝弘: 電力系統事故時復旧支援, 人工知能学会誌, Vol.7, No.4, pp.582-586 (1992).

(平成 6 年 5 月 23 日受付)

(平成 7 年 10 月 5 日採録)



大川 剛直 (正会員)

昭和 38 年生。昭和 61 年大阪大学工学部通信工学科卒業。昭和 63 年同大学院博士前期課程修了。平成 2 年大阪大学工学部通信工学科助手。現在、大阪大学工学部情報システム工学科講師。博士(工学)。知識獲得、定性シミュレーションなどの研究に従事。IEEE などの会員。



馬場口 登 (正会員)

昭和 32 年生。昭和 54 年大阪大学工学部通信工学科卒業。昭和 56 年同大学院前期課程修了。昭和 57 年愛媛大学工学部助手。大阪大学工学部助手、講師を経て、現在、大阪大学産業科学研究所助教授。工学博士。人工知能、パターン認識、画像処理の研究に従事。IEEE、電子情報通信学会、計測自動制御学会、人工知能学会各会員。