

### 5.3 類推による学習

類推では、いくつかの与えられた対象間に類比，すなわち類似性を発見し，その類比を用いて一方の対象で成立する事実，関係などの知識を他方のそれに変換する．類推により，問題解決の手掛りを得たり，未知の事実を予測推定することができる．ここでは，ルールの分解と合成による学習法を用いて類推が行なえることを示す．

#### 5.3.1 類推研究の概要

類推の研究は Winston のシステム [Winston 80] [Winston 82] [Winston 86] がきっかけとなって盛んになった [松原 86]．Winston の作成したシステムでは，入力としてまず Macbeth のプロットが英文で与えられる．次に異なった状況での事件について質問すると，システムは与えられたプロットの因果関係との類推により事件の説明を試みる．類推を行なうためには，まずプロットと質問の事件の状況との照合を行なう．照合は，ありうる組み合わせすべてについて試みられ，得点づけによりもっともらしい対応付けが選択される．次に，その対応付けにしたがって類推を行なう．Winston の方法は因果関係に注目したもので，似た状況が似た結末を生じるという仮定にしたがっている．

Winston の研究に触発され，Gentner による構造写像 (structure-mapping) 理論 [Gentner 83]，Kedar-Cabelli による目的志向の類推 (purpose-directed analogy) [Kedar-Cabelli 85] などが提案された．また，Burstein により，プログラミング言語 Basic における代入文の意味を，箱の概念からの類推により対話的に学習するシステム CARL が作成されている [Burstein 86]．Winston の研究は細部にこだわらない直感的な議論に基づいているが，原口はこれを部分同一の概念により精密に理論化し，さらに演繹と類推を使い分けながら連続的に推論を行なうシステムを作成した [Haraguchi 84] [原口 85] [原口 86a] [原口 86b] [原口 86c]．

以上の研究はそれぞれ興味深い理論に基づいて行なわれているが、基礎となるメカニズムの裏付けは全くない。ここでは、基礎となるメカニズムとして分散処理モデルを仮定した場合、類推が可能であることを示す。すなわち、ルールの分解と合成による類推の方法と、それによる実験結果について述べる。

### 5.3.2 幾何学的類推の問題

図5.13 は知能テストなどに用いられる幾何学的類推の問題[Winston 84]である。この問題では、「AがBに対するのは、CがXに対するのと同じである。」という記述に最もあうXを、類推する。

類推の方法を図式化すると図5.14 に示すようになり、類推T'を、AとCの類比と変換Tより求めることになる。すなわち、AからCへの変換ルール  $A \rightarrow C$  を分解することによって類比Rを表すルール集合を求め、それを用いて類推T'を推論すればよい。

まず、類比をおこなうため、図5.13 を次のように表現する。

```
A : (:relation: above (:object: triangle l) (:object: square m))
B : (:relation: left  (:object: triangle l) (:object: square m))
C : (:relation: above (:object: dot x)      (:object: circle y))
```

まず、Aを左辺にもち、Cを右辺にもつ例 Rinit を生成する。

```
Rinit: (AB (:relation: above (:object: triangle l)
                             (:object: square m)))
      → (CX (:relation: above (:object: dot x)
                             (:object: circle y))).
```

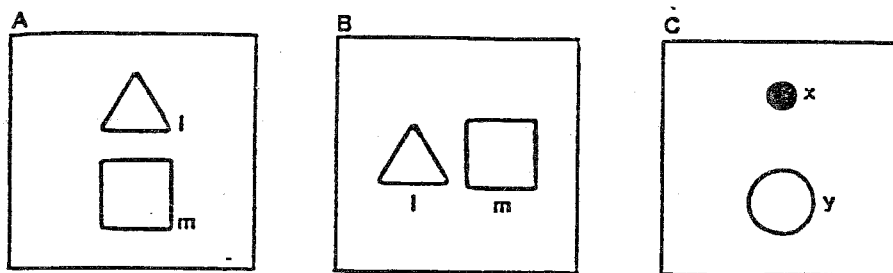


図 5.13 幾何学的類推の問題

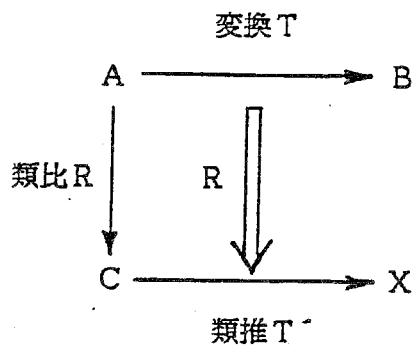


図 5.14 類推の図式

ここで、AB および CX は A と C を区別するためのタグである。上のルールを分解し、類比 R を表すルール集合を求める。

まず、例 Rinit を 図 5.15 に示すような基本ルールに分解し、図 5.16 に示すような構成グラフを生成する。図 5.15 は、基本ルールの一部を示したものである。等価変換ルールは等価な変換だけを行なうルールであり、Rinit の左辺と右辺が等しいときには、構成グラフには等価変換ルールだけが含まれることになる。抽象化ルールは図形の要素および関係を捨象することにより、Rinit の左辺を抽象化する。具象化ルールは、同様にして、Rinit の右辺の抽象化を行なう。すなわち、左辺から右辺を導くという立場でみれば、具象化を行なうルールとなる。等価変換ルールの適用コストを 0 とし、抽象化および具象化ルールの適用コストを 1 としているので、これらのルールによる最小コストのパスにより、A と C の最大類比が表現される。すなわち、A の三角形および四角形と B の点および円における「図形の形」を捨象して、単なる物体であるとみなすことにより、A と C の類比をとることができる。この構成グラフを、A から C が決定的に導かれるように分割し、各部分グラフ中のルールを合成する。その結果、次のような類比 R を表わすルール集合が得られる。

encode-relation:

(AB (:relation: \*relation \*x \*y))

⇒ (:relation: \*relation (AB \*x) (AB \*y)) cost: 0.

decode-relation:

(:relation: \*relation (CX \*x) (CX \*y))

⇒ (CX (:relation: \*relation \*x \*y)) cost: 0.

triangle-dot:

(AB (:object: triangle \*x)) ⇒ (CX (:object: dot \*x1)) cost: 0.

square-circle:

(AB (:object: square \*x)) ⇒ (CX (:object: circle \*x1)) cost: 0.

a) 等価変換ルール

encode-relation:

(AB (:relation: \*relation \*x \*y))

⇒ (:relation: \*relation (AB \*x) (AB \*y)) cost:0.

encode-object:

(AB (:object: \*object \*x))

⇒ (:object: \*object) cost: 0.

decode-relation:

(:relation: \*relation (CX \*x) (CX \*y))

⇐ (CX (:relation: \*relation \*x \*y)) cost: 0.

decode-object:

(:object: \*object)

⇐ (CX (:object: \*object \*x)) cost: 0.

b) 抽象化および具象化ルール

abstract-triangle:

(AB (:object: triangle \*x))

⇒ (:object: arbitrary) cost: 1.

instantiate-triangle:

(:object: arbitrary)

⇐ (CX (:object: triangle \*x)) cost: 1.

図5.15 類比のための基本ルール(一部分)

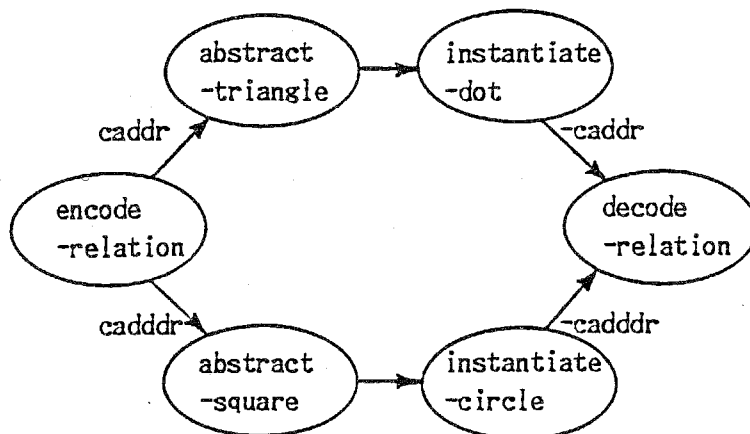


図5.16 Rinit の構成グラフ

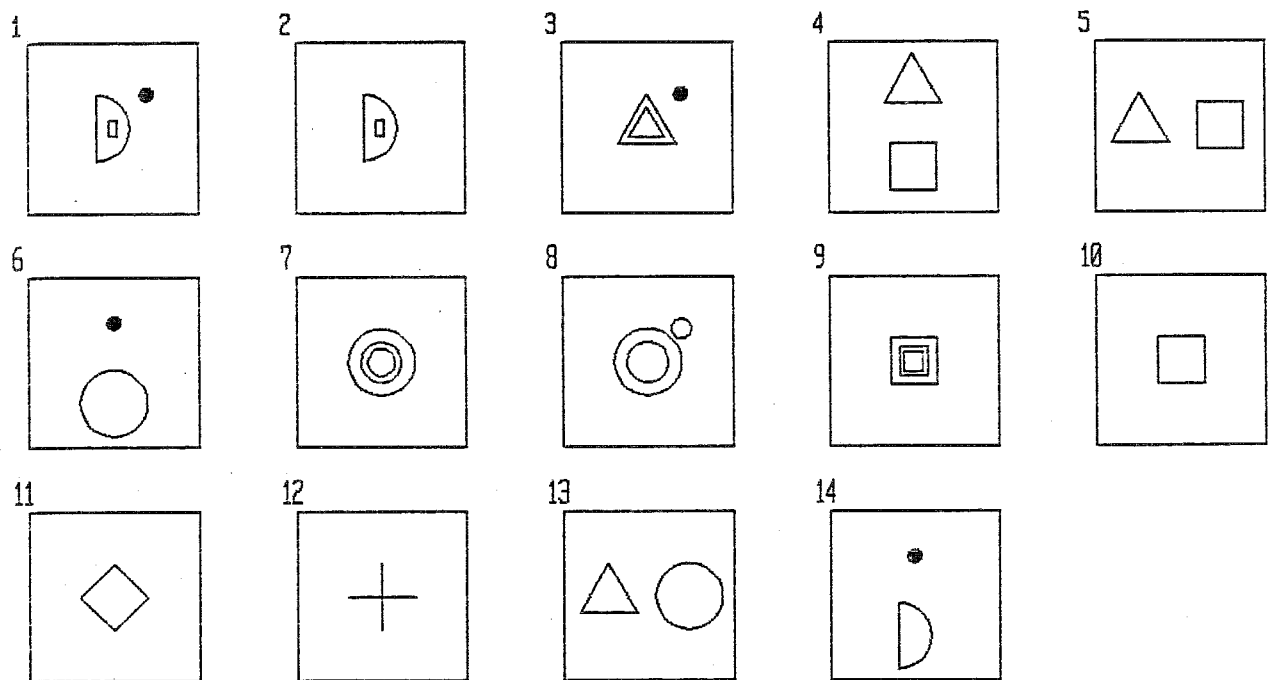
以上のようにして得られたルール集合を変換 T に適用することにより類推 T' が次のように求まる。

```
T' : (CX (:relation: above (:object: dot *x4)
                             (:object: circle *x3)))
      → (CX (:relation: left (:object: dot *x2)
                              (:object: circle *x1))).
```

Evans の幾何学的類似問題のプログラム[Winston 84]では、類推結果を与えられた選択肢から選んだが、上の類推では直接解答を生成していることに注意されたい。

幾何学的類似問題は視覚に訴えるので格好のデモプログラムになっている。このデモでは、人間が入力した幾何学的類似問題に対し、もっともらしい解答を生成する。まず、図 5.17 に示すようなメニューが表示され、このメニューのなかから A, B, および C を選ぶ。すると、選んだ図形が 図 5.18 のように表示され、システムが類推の結果生成した図形を X として表示する。デモプログラムのメニューの各図形に対する知識表現および使用した基本ルールの詳細については、付録 III を参照されたい。

デモプログラムが解いた問題の例と、解くのに要した所要時間を 図 5.19 に示す。(1)の例では、半円と長方形と点からなる図形に対し、点のない図形が対応している。このとき、C として三角形二つと点からなる図形を与えると、類推によって点のない図形が生成される。(2)の例では、三角形と四角形が上下に並んだ図形に対し、三角形と四角形が左右に並んだ図形が対応している。このとき、C として点と円が上下に並んだ図形を与えると、点と円が左右に並んだ図形が生成される。(3)の例では、円が三つ入れ子になった図形に対し、一つの円が外に出た図形が対応している。このとき、C として四角が三つ入れ子になった図形を与えると、四角が一つ外に出た図形が生成される。(4)



私は、類推をするプログラムです。  
「AがBに対するのは、CがXに対するのと同じである。」  
にあてはまるXを推論します。

上の中から、A、B、Cを選んで下さい。

A: 4  
B: 10  
C: 13

図5.17 類推のデモプログラムメニュー

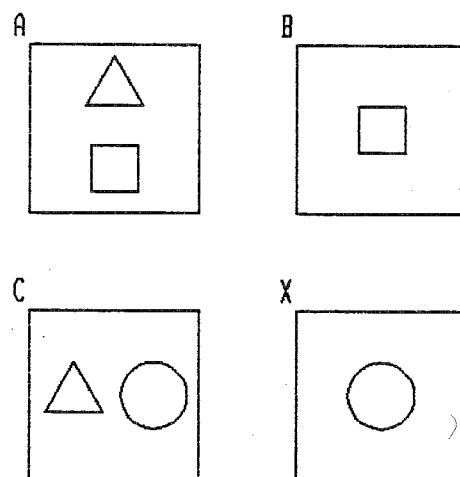
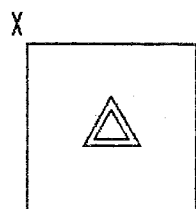
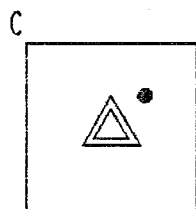
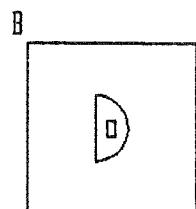
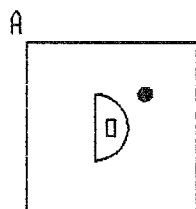


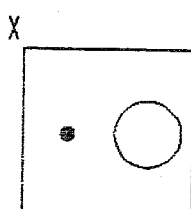
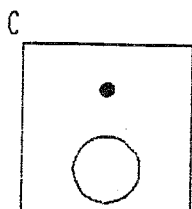
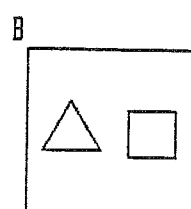
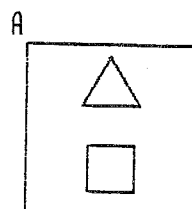
図5.18 選択した問題

(1)



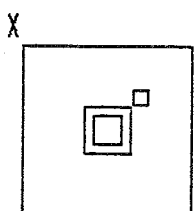
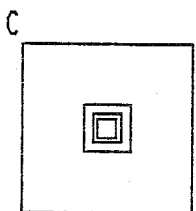
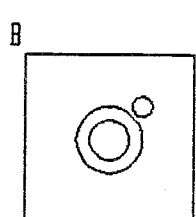
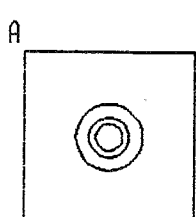
26.3秒

(2)



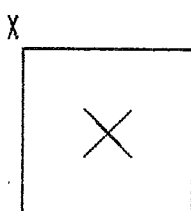
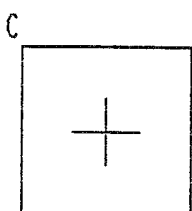
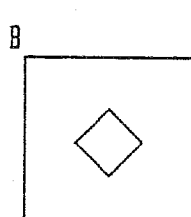
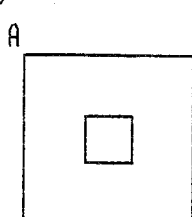
13.9秒

(3)



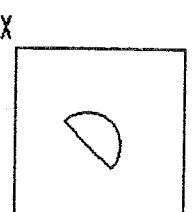
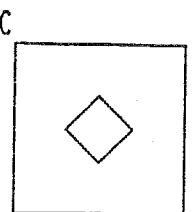
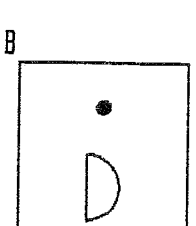
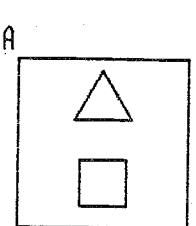
33.9秒

(4)



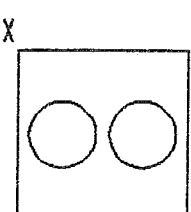
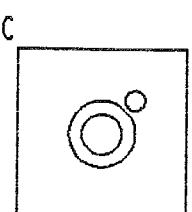
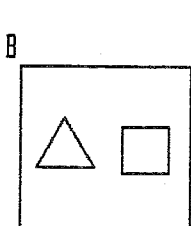
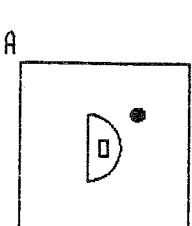
5.7秒

(5)



13.6秒

(6)



26.1秒

図5.19 類推の実験結果



の例では、正方形に対し正方形を45度回転した図形が対応している。このとき、Cとして十字形を与えると、十字形を45度回転した図形が与えられる。(5)の例では、三角形と四角形からなる図形に対し、点と半円からなる図形が対応している。このとき、45度回転した四角形を与えると、半円が45度回転した図形が生成される。

以上の例は人間が見ればすぐに結果が判る例であるが、デモにおいては図5.17のメニューの中からA, B, Cを自由に選ばせるので、ほとんど無関係な図形が問題として与えられる場合がある。(6)は、そのような問題の例である。この場合プログラムは、三個の図形と、左右に並んだ二つの図形を対応させていると考えて、三個の円からなる図形に対し円が二つ左右に並んだ図形を類推した。納得できる解答であると思われる。

### 5.3.3 類推のまとめ

ルールの分解と合成により、類推を行なう方法を述べた。ここで述べた例では知識表現がS式であり表現力が不足しているが、方法としては十分に一般性がある。4章で述べたようにNatに關係表現を導入すれば、この方法により他分野での応用も期待できるであろう。

#### 5.4 まとめ

ルールの分解と合成による学習法を用いて、演繹、帰納、および類推による学習が行なえることを示した。ルールの分解による一般化は、プロダクションシステムによる探索とルールの合成によって行なわれるが、このうち、前者は古典的な認知のモデルであり、後者はニューロンにおけるシナプスの形成のモデルになっている。したがって、これらの認知科学的モデルに基づいた単純な機構により、多様な学習が可能であることが示された。

また、帰納的推論において明らかとなったように、本学習法における基本ルールは、誤りを含んでいるのが普通であり、論理学における公理と異なり無矛盾性等は要求されない。すなわち、基本ルールに「正しさ」を要求しないことにより、学習が行なえるようになるのである。