

人工知能 課題番号 08 「幾何図形類推問題」

工学部電子情報工学科 03-175001 浅井明里

2017 年 11 月 7 日

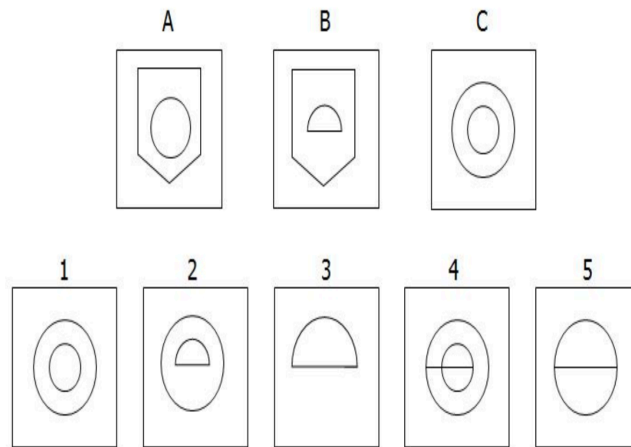


図 1 幾何類推問題の例

## 1 Evans のプログラム ANALOGY がどのように幾何図形類推問題を解くことができたか

類推 (Analogy) とは、ある物事と別の物事の間何らかの類似に基づいて一方がある性質を持つ場合に他方もそれと同じ性質を持つであろうと推理することであり、Evans の ANALOGY プログラムはこの幾何の類推を行う古典的なプログラムである。ANALOGY では以下のようなステップで幾何類推問題を解く。

1. それぞれの問題の図を複数の”Object(sub figure)”に分解し、その関係を記号表現に直す。
2. 記号的な表現のマッチングを行う。
3. ある幾何図形からある幾何図形に変換するルールを生成する。

この項では、この ANALOGY プログラムがどのようにして幾何類推問題を解いていたのかを以下の図 1 の例を元に紹介していきたい。

### 1.1 図 A, B, C の類似点と相違点を記述するプログラムでは、各図をどのように記述すれば良いか

まず、図 A、B、C をそれぞれ記号表現で記述する。記述の仕方については ANALOGY を提案した論文及び参考書より、図に含まれるオブジェクトを列挙する CONSISTS-OF、またそれぞれのオブジェクトの関係性を記述する RELATIONS で表現し、また RELATIONS については INSIDE, ABOVE の二種類で表現し、(INSIDE SubFig1 SubFig2) ならば SubFig1 が SubFig2 に含まれていることを、(ABOVE SubFig1 SubFig2) ならば SubFig1 は SubFig2 の上方にあることを表すとする。またそれぞれのオブジェクトに以下の図 2 のように識別のための名前をつけることとする。

```
(FIG A
  (CONSISTS-OF P1 P2)
  (RELATIONS (INSIDE P1 P2)))
```

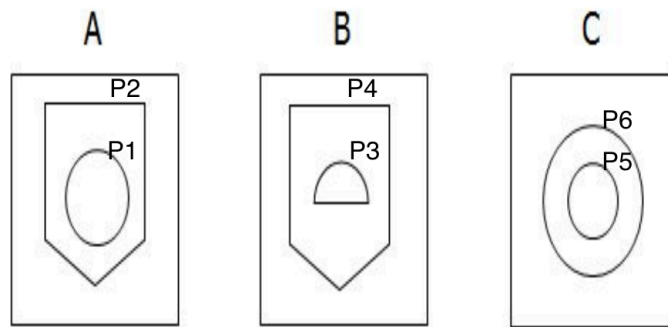


図2 幾何類推問題の例

```
(FIG B
  (CONSISTS-OF P3 P4)
  (RELATIONS (INSIDE P3 P4)))
(FIG C
  (CONSISTS-OF P5 P6)
  (RELATIONS (INSIDE P5 P6)))
```

次に、A、B 間の類似性、A と C の間の類似性を記号から推定する。ある Figure1 から Figure2 に類似性は (SIM-Figure1-Figure2) と表し、そのうち特に Figure1 に含まれる SubFigure1-1 から Figure2 に含まれる SubFigure2-1 に変換が可能であるとき、変換情報は (TRANS a b c d) で表現される。このとき a,b,c,d はそれぞれ以下の変換情報を示している。

- a, d : 対象変換の種類。
  - K : 鏡像
  - V : 垂直
  - H : 水平
- b : 相似変換の拡大率
- c : 回転変換の角度

```
(SIM-A_B (SIM P2 P4 : (TRANS K 1.0 0 K)))
(SIM-A_C (SIM P1 P5 : (TRANS K 1.0 0 K)))
```

以上より、A と B において P2 と P4 が、A と C において P1 と P5 が類似であることが導かれる。

## 1.2 図Aを図Bに変える規則の記号的な記述を与え、この規則を上で与えた記述からどのように機械的につくることができるかを説明せよ。

前項で求めた類似性の記号表現より、図Aを図Bに変える規則の記号的な表現を求める。変形の手順については以下のように記述する。

- **REMOVE SubFigure1 WITH (ConditionsA)**  
ConditionA に合致する SubFigure1 を取り除く。

- **MATCH SubFigure2 FROM (ConditionsB) TO (ConditionsC) WITH (TRANS a b c d)**  
SubFigure2 を変換情報 TRANS を維持したまま、条件 ConditionB から条件 ConditionC を満たすように変更する。
- **ADD NewFigure1 WITH (ConditionsA)**  
NewFigure1 を条件 ConditionsA を充しながら追加する。

この規則を与えた上で、前述の図 A と B の類似関係から機械的に導くと、変更規則の記号的な記述は次のようになる。これは以下のような手順で機械的に求めることができる。

オブジェクト P2 と P4 は類似しており、追加も削除も必要なくまた図形的にも全く同じ像であるため変換規則の適用も必要ない。次に P1 と P3 であるが、この二つの図形に類似性はなく (P1 を半分に分割したものが P2 と捉えることもできるが、この問題ではこの二つの図形を類似性のない独立したものであると捉えることとする)、図 A から図 B の変換には P1 を除去し、P3 を追加する必要があることがわかる。P1 を特定する条件を WITH 以下に記述し、また (SIM OBJ3 ...) は図 A と図 B の図形の種別を比較するための方法を示している。最後に新たな図形 P3 を追加する部分では、追加する際の条件 (この場合は既存するもう一方の図形の内部に配置する) と共に図形を追加する旨を記述する。最終的に変換規則は以下のように表現できる。

```
((REMOVE x1 WITH ((INSIDE x1 x2)
                    (SIM OBJ3 x1 (TRANS K 1.0 0 K)))
(ADD x2 WITH ((INSIDE x2 x3))))
```

以上の手順により、図形 A から図形 B への変換規則が機械的に示された。

### 1.3 自分の作った規則を図Cに対して自分の作った記述に適用せよ。その結果どのような記述が得られ、それはどのような図であることを説明せよ。この図と図Cとの間の類似点を記述せよ。

以上の変換規則を、図 C に適用した場合以下のように記述でき、すなわち図 C に含まれる二つの図形のうち、一方のより大きな図形に含まれるもう一方の図形を除去し、新たな図形を残っている図形の内部に追加すればいいということがわかる。

```
((REMOVE P5 WITH ((INSIDE P5 P6)
                    (SIM OBJ3 x1 (TRANS K 1.0 0 K)))
(ADD Px WITH ((INSIDE Px x6))))
```

新たな図形 X 及び C と X の類似性は次のような記号的表現記述により表すことができ、結果として ANALOGY プログラムに図 1 のような問題を与えたとき、得られる答えは図 1 の「2」となることがわかる。

```
(FIG X
  (CONSISTS-OF P7 P8)
  (RELATIONS (INSIDE P7 P8)))
(SIM-C_X (SIM P6 P8 : (TRANS K 1.0 0 K)))
```

### 1.4 図Cの記述方法を変えて、同じ規則をその記述に適用すると、前とは違った結果になるようにせよ。

次に、図 C の記述方式を以下のように変換する。

```
(FIG C
  (CONSISTS-OF P6 P5)
  (RELATIONS (INSIDE P5 P6)))
```

この図 C の記述に基づいて、図 A と図 C の類似性を記号的に表現する場合、以下の通りになる。

```
(SIM-A_C (SIM P1 P6 : (TRANS K 0.8 0 K)))
```

すなわち、図 C の表現方法を変化させたことで図 A とのオブジェクトの対応関係が変化し、その結果図 A と図 C の類似性の記述に変化が生じた。図 A の P1 に対応する、すなわち返還後除去されることとなるオブジェクトは P6 となり、P5 は除去されることなく残存し、新たなオブジェクトはこの P5 の内部に存在することになる。

```
(FIG X
  (CONSISTS-OF P7 P8)
  (RELATIONS (INSIDE P7 P8)))
(SIM-C_X (SIM P5 P8 : (TRANS K 1.0 0 K)))
```

結果として返還後に得られる図は前項の回答と一回り小さい図形となることがわかる。

## 1.5 Evans のプログラム ANALOGY では解くことができないであろう幾何図形類推問題の例を示し、なぜ解けないのかを説明せよ。

Evans のプログラムでは解くことのできない問題の例をいくつかあげたいと思う。

Evans のプログラムにおいて、図に含まれるそれぞれのオブジェクトは二つの図を比較するとき図に含まれるオブジェクト同士が互いに一対一で対応する必要がある。しかし、例えば「図 A において二つのオブジェクトが与えられており、それらを結合すると図 B となる」というような変換規則であった場合、ANALOGY による幾何類推では、図 A 及び図 B の間のそれぞれのオブジェクト間に一対一関係を構築することができないため、正しく類推をすることができない。以下で例を示す。

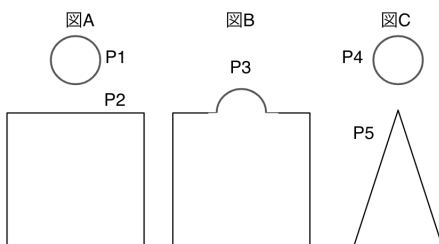


図 3 正しく類推ができない問題の例

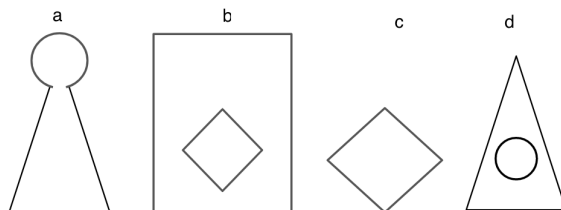


図 4 ANALOGY で正しく類推ができない問題の選択肢

図 3 のような例が与えられれば、人間であれば、「図 A には丸と四角形が存在しており、図 B に変換するためにはこの二つの図形を結合させれば良い。よって、図 C に同様の変換規則を適用すれば、三角形と丸という二つのオブジェクトが結合し、図 4 の (a) となる」と類推できるであろう。ANALOGY でこの問題を解こうとして場合それぞれの図及びそれらの類似性は以下のように表現される。

```
(FIG A (CONSISTS-OF P1 P2)
  (RELATIONS (ABOVE P1 P2)))
```

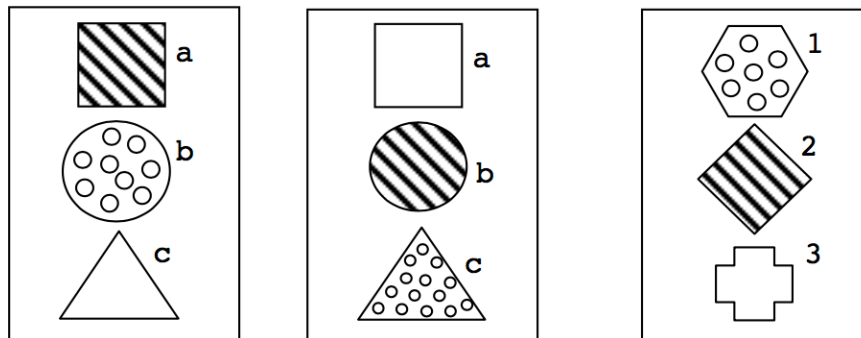


図5 幾何類推問題の例

```
(FIG B (CONSISTS-OF P3))
(FIG C (CONSISTS-OF P4 P5))
  (RELATIONS (ABOVE P4 P5)))

(SIM-A_B ())
(SIM-A_C
  (SIM P1 P4 : (TRANS K 1.0 0 K)))
```

ANALOGY では図に含まれるオブジェクトを分離する際、重なっている図形については単一閉曲線の図形になるように分離すれば良いとしたが、この方式で図 A, B, C を見るとき、A, C についてはそれぞれ丸と正方形、丸と三角形という二つのオブジェクトに分離が可能だが、B については正方形に丸が結合された図形であっても、これは P3 という単一のオブジェクトとして考えられることとなる。その結果、図形 A と B の間の類似性を見つけようにも A の二つのオブジェクトと類似する図形は存在しないため見つけることができない。故に、A から B への変換規則は次の通りとなる。

```
(REMOVE x1 WITH ((ABOVE x1 x2)
  (SIM OB3 x1 (TRANS K 1.0 0 K)))
(REMOVE x2 WITH ((ABOVE x3 x2)
  (SIM OB3 x1 (TRANS K 1.0 0 K)))
(ADD x3 ()))
```

すなわち、図に存在する二つのオブジェクトを除去し、新しいオブジェクトを追加するというのが変換規則となってしまう、この場合類推の結果提示される回答は (c) となるがこれは誤った類推結果である。またこの問題は結局のところ ANALOGY プログラムにとって図 A と B が全く異なる図形として解釈された場合、ANALOGY は正しい類推結果を出すことができないことも暗示している。

また、オブジェクトが複数のアトリビュート (形、位置関係、パターン、色等) を持つ問題も ANALOGY プログラムでは正しい類推結果を求めることが難しいのではないかと推測する。例として図 5 に示された問題について考えたい。図形を記号的表現に落とし込むに当たって、新たに PATTERN というアトリビュートを導入したい。PATTERN には STRIPE, DOT, PLAIN という 3 種類の柄のいずれかが全ての図形に対して与えられる。

```
(FIG A (CONSISTS-OF P1 P2 P3)
  (RELATIONS (ABOVE P1 P2)(ABOVE P2 P3))
  (PATTERN (STRIPE P1)(DOT P2)(PLAIN P3))
```



図 6 幾何類推問題の例

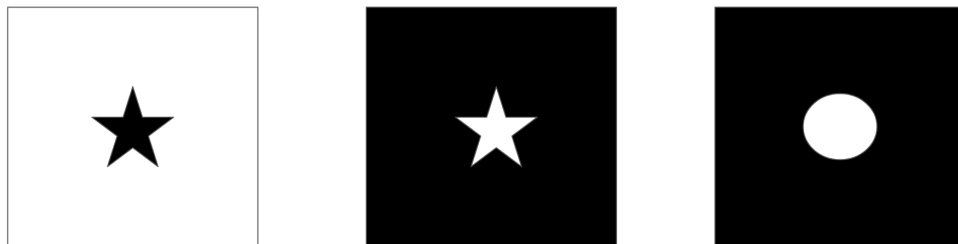


図 7 幾何類推問題の例

```
(FIG B (CONSISTS-OF P4 P5 P6)
  (RELATIONS (ABOVE P4 P5)(ABOVE P5 P6))
  (PATTERN (PLAIN P1)(STRIPE P2)(DOT P3))

(FIG C (CONSISTS-OF P7 P8 P9)
  (RELATIONS (ABOVE P7 P8)(ABOVE P8 P9))
  (PATTERN (DOT P1)(STRIPE P2)(PLAIN P3))
```

ここで問題となるのは、オブジェクト同士の一対一対応関係を求めるにあたり RELATIONS と PATTERN という二つのアトリビュートのうち、どちらが優先されるべきかが明示されていないため RELATIONS を優先させた類推結果と PATTERN を優先させた類推結果の複数種類の結果が導き出されることとなる。

他に ANALOGY が解くことのできない類推問題として、「数え上げ」「背景の反転」等、事前に機能として想定されていれば類推可能だが、そうでなければ解くことのできない問題が想定される。図 6 で示された例では「図 A にあるオブジェクトの数が図 B で数字で示される」という変換規則が成立し、これを一番右の図に適用すると、この図は二つのオブジェクトを持つため変換後は数字の「2」が中央に配置された図になるべきだと予測できる。しかし前述の記号的表現及び ANALOGY の機能では、こういったオブジェクトの数を数え、それによって追加するオブジェクトを決定する機能はなくこの問題を解くことができない。

また図 7 の問題に存在する変換規則は「図のオブジェクトの色と背景色を反転させる」というものと推測できるが上述の記号的表現ではこの「色の反転」に相当する表現は定義されておらず、既存の色やパターンのみで判断することも前述の複数マッチングの観点より困難であり、やはり正しい類推を行うことのできない可能性がある。

## 1.6 Evans のプログラム ANALOGY と I Q テストに用いられるような幾何図形類推問題に関する以下の命題について、上で示した例などを利用して論ぜよ。

### 1.6.1 幾何学的図形類推問題を解く知的行為の基本的な要素は、純粋に機械的なものであることを示した。

幾何学的図形類推問題を解く知的行為の基本的な要素は純粋に機械的なものではないと私は考える。類推を行う際、まず最初に図形及びそれらの類似性を記号的表現で示す必要がある。与えられた図形画像を認識し、ある程度機械的にこの作業を行うことは現在の技術では可能かもしれないが、仮にそうであった場合にもいくつかの問題点が存在していると考ええる。

第一の問題はある図形を複数のオブジェクトに分解し、図形同士の関係性も合わせて記号的表現に落とし込む段階にあると考える。図を複数のオブジェクトに分解する際に、「単一閉曲線」で図形を区切ることが可能かどうか与えられた問題に依存しており、例えば前述のようなオブジェクト同士が結合してしまった図の場合、人間ならば隠れた線を推測し無意識にその図形が丸と四角の結合したものだとして認識できるが類推プログラムの場合明示的にこのルールも追加しなくては正しくオブジェクトに分解することができなくなる。ただこのとき、どの場合では単一閉曲線を二つのオブジェクトが結合したものと認識するのか、例えば菱形は三角形が二つ結合したものと考えるべきなのか等多くの条件の設定が必要となり、結果的に人手である程度オブジェクトの分解についてのルールを与える必要があり機械的に図形を記号的表現に正しく変換可能だとは考えられない。

第二の問題は複数のアトリビュートを持つ図形の類似性の記述におけるアトリビュートの優先順位に関する問題である。図5の問題のような、図に含まれるオブジェクトが複数のアトリビュートを持つ際にこのアトリビュートの優先順位をどう付けて図同士の対応関係を求めるか、図形の変換を行うかにより答えが変わってしまう。Bohan(2000) らはこの問題に対し、オブジェクトのペアに対して適用される local アトリビュート変換と全てのオブジェクトに対して適用される global アトリビュート変換という異なるアトリビュート変換を用い、この問題の解決を試みた。ただ解決法についても、アトリビュートの優先順位付についても人の手により明確に「基準」を与える必要があり、完全に機械的に行うことができない。

以上より、幾何学的図形類推問題を解く知的行為の基本的な要素は純粋に機械的なものではないと私は考える。

### 1.6.2 幾何学的図形類推問題を解くのに必要な知的な能力は、もともになる記号的記述体系をどのように選ぶかということにのみ存在する

幾何学的図形類推問題を解くのに必要な知的な能力は、もともになる記号的記述体系をどのように選ぶかということにのみ存在するという命題は偽りであると私は考える。仮に元となる記号的記述体系を決定できたとしても、幾何類推のためにはそれ以外にも機械的に解くことのできない段階が存在するためである。記号的記述体系以外に必要なだと私が考える知的能力は以下の二つがある。

まず、あらかじめ定められた記号的記述体系で表現し得ないような問題に直面した時、現在自分の持つ機能、記号的表現だけでは十分にその問題に対し正しい幾何類推を行うことができないと判断できる能力である。数え上げ問題、反転問題等は問題中の図自体を記号的表現に落とし込むことは既に定義された記号的表現で十分に可能である。しかし一方で、数え上げ問題を「全てのオブジェクトを除去し、必ず1個のオブジェクトを中央に配置する」と解釈してしまったり、反転を「グローバルなトリビュートの変換」と解釈してしまったりした場合、おそらく問題で想定された解を導くことは難しい。ここで必要なのは問題に直面する度に、まず自身の持つ記号的表現を適用してみるのではなく、そもそもこの問題は自身の今現在持つアプローチで正しく回答



ができるのかを判断する知的能力が必要である。

次に与えられた問題に既存の機能、記号的表現では十分に対応できないと判断した場合、自ら新たに必要な記述体系を生み出し類推をすることである能力である。そこで既に獲得している記号的記述体系等を参考にしつつ新たな記述体系、機能を生み出していく必要があり、この場面においても知性は必要である。

この点に対して、全ての問題を解くために必要な機能、記号的表現をあらかじめ全てプログラムしておけばいいのではないかという反駁も想定できるが、どのような幾何類推問題が聞かれうるか全てあらかじめ想定仕切することは不可能であり、また新たに生み出される問題に対して十分に対応できるものでなければ幾何類推問題を適切に解くことのできるプログラムだということはできない点で全ての問題を想定するという前提が実現性のないものであるためこの反駁は不十分である。

以上より、幾何学的図形類推問題を解くのに必要な知的な能力は、もともになる記号的記述体系をどのように選ぶかというところのみ存在するという命題は偽りであると私は考える。

## 参考文献

- [1] Thomas G. Evans, "A HEURISTIC PROGRAM TO SOLVE GEOMETRIC-ANALOGY PROBLEMS", Air Force Cambridge Research, Laboratories (OAR) Bedford, Massachusetts, 1968.
- [2] 伊庭齊志, 『人工知能と人工生命の基礎』, オーム社, 2013.
- [3] A. Bohan, D. O'Donoghue, "LUDI: A Model for Geometric Analogies using Attribute Matching", Dept. Computer Science, NUI Maynooth, Co. Kildare, Ireland, 2000.