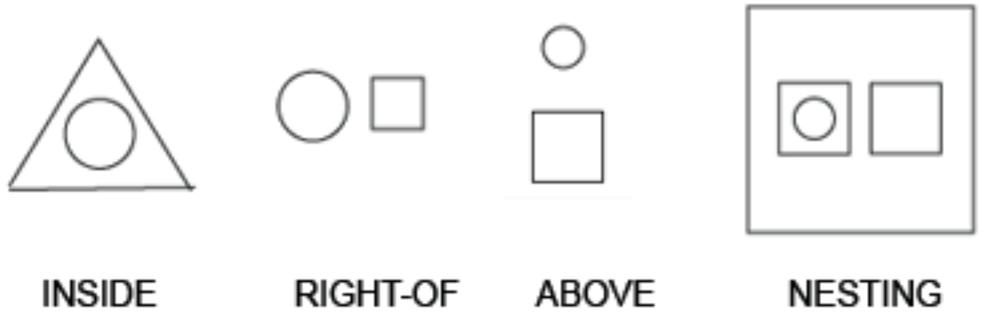
Angela Gabereau
4867815
COMP 472 –Intro to A.I.
Novermber 9, 2007

GEOMETRIC ANALOGY REASONER

KNOWLEDGE REPRESENTATION



The knowledge representation I chose represents geometric shapes as triples contains two shapes and a relation between the two shapes. These triples can be nested. A triple can serve as a shape in a triple. This representation allows for any type of relation between 2 shapes. Below are some examples:

INSIDE: (inside triangle circle)
RIGHT-OF: (right-of circle square)
ABOVE: (above square circle)
NESTING: (inside square (right-of (inside square circle) square))

ASSUMPTIONS

The assumptions for this analogy reasoner are:

1. All analogy problems of the form “A is to B as C is to D”
2. The shapes in segment B are a subset of the shapes in segment A. (Likewise for segment D and segment C). Therefore this reasoner cannot solve problems where segment B contains a shape that is not in segment A or segment D contains a shape that is not in segment C.
3. There is a pattern in segment A that is analogous to a pattern in segment C. (Likewise for segment B and segment D).

ANALOGY HEURISTIC

The steps performed by the reasoner to solve the analogy are:

1. Place all the shapes in segment A on the list mappingAB. There is a parallel list called shapes which contains ‘(s1 s2 s3 s4 s5 s6 s7 s8 s9). Once all shapes from A have been placed on mappingAB they are associated with a value on shapes. For example, if mappingAB contained ‘(circle square triangle circle) then s1=circle, s2=square, s3=triangle and s4=circle.
2. Translate segment B into a pattern using the lists mapping AB and shapes. For example, if segment B = (inside circle (inside square (right-of circle triangle))), using the mapping from 1 we get patternB = (inside s1 (inside s2 (right-of s4 s3)))
3. Place all the shapes in segment C on listing mappingCD. Each shape is associated with value on list shapes as in 1.
4. Using the mappingCD, shapes and patternB create patternD by substituting the values in patternB for their associated values in mappingCD. For example if mappingCD=’(square circle triangle square) and patternB=(inside s1 (inside s2 (right-of s4 s3))) the patternD will be (inside square (inside circle (right-of square triangle)))
5. Find if any of the answer segments (D1, D2, D3 or D4) matches patternD. If so output result. Otherwise this reasoner cannot solve the analogy.

This reasoner functions by creating a pattern from A and B and then substituting the values in C to infer D. It can only solve problems for which there is the same pattern in AB as CD but with different shapes. It can only solve problem for which the shapes in the second segment of the “is to” is a subset of the shapes in the first segement. This reasoner can only solve problems for which the shapes within can be represented as relations between 2 shapes in one segment.

INSTRUCTIONS

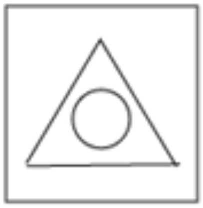
To run the reasoner in gambit type (load “geometric_analogy.scm”).
This will run the reasoner with Test Case 1. To change test cases open up “test_cases.scm” and choose a test case and then replace the test case in geometric_analogy.scm.

NOTE: In the function get_analogy:

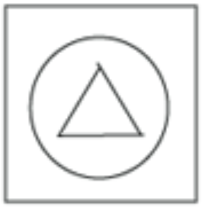
```
(define get_analogy
  (lambda ()
    (cond ((equal? patternD D1) (display_analogy D1 "D1"))
          ((equal? patternD D2) (display_analogy D2 "D2"))
          ((equal? patternD D3) (display_analogy D3 "D3"))
          ;((equal? patternD D4) (display_analogy D4 "D4"))
          (else (display "Sorry, could not solve this analogy"))
          )
    )
  )
```

The line ((equal? patternD D4) (display_analogy D4 "D4")) must be commented out if the analogy has 3 possible answers. This is the case in test cases 3 4 5 and 6.

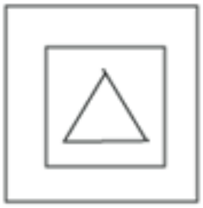
TEST CASE 1



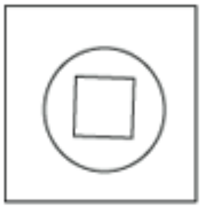
A



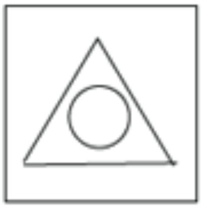
B



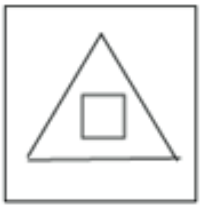
C



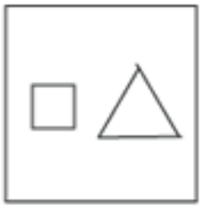
D1



D2



D3



D4

From Artificial Intelligence, Luger

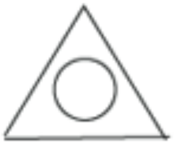
REPRESENTATION

```
(define A '(inside square (inside triangle circle)))
(define B '(inside square (inside circle triangle)))
(define C '(inside square (inside square triangle)))

(define D1 '(inside square (inside circle square)))
(define D2 '(inside square (inside triangle circle)))
(define D3 '(inside square (inside triangle square)))
(define D4 '(inside square (right-of square triangle)))
```

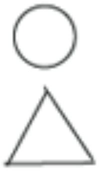
RESULT

```
GEOMETRIC ANALOGY
----------------------------------
A IS TO B
AS
C IS TO D3

(inside square (inside triangle circle)) IS TO (inside square (inside circle triangle))
AS
(inside square (inside square triangle)) IS TO (inside square (inside triangle square))
```

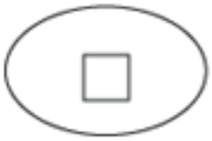
TEST CASE 2



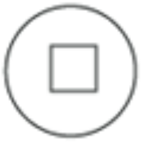
A



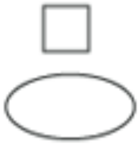
B



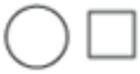
C



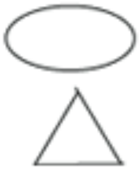
D1



D2



D3



D4

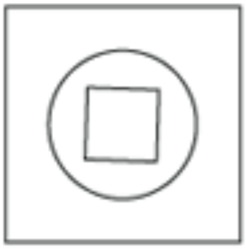
REPRESENTATION

```
(define A '(inside triangle circle))
(define B '(above triangle circle))
(define C '(inside oval square))

(define D1 '(inside circle square))
(define D2 '(above oval square))
(define D3 '(right-of circle square))
(define D4 '(above triangle oval))
```

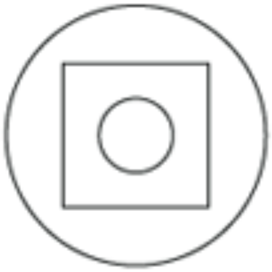
RESULT

```
GEOMETRIC ANALOGY
---------------------------------
A IS TO B
AS
C IS TO D2

(inside triangle circle) IS TO (above triangle circle)
AS
(inside oval square) IS TO (above oval square)
```

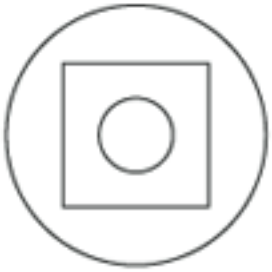
TEST CASE 3



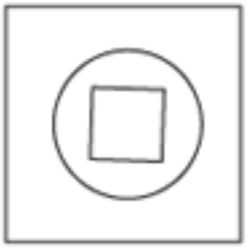
A



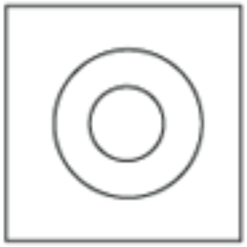
B



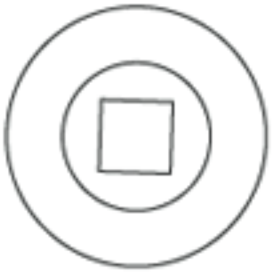
C



D1



D2



D3

From Sabine Bergler

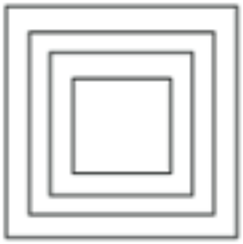
REPRESENTATION

```
(define A '(inside square (inside circle square)))
(define B '(inside circle (inside square circle)))
(define C '(inside circle (inside square circle)))

(define D1 '(inside square (inside circle square)))
(define D2 '(inside square (inside circle circle)))
(define D3 '(inside circle (inside circle square)))
```

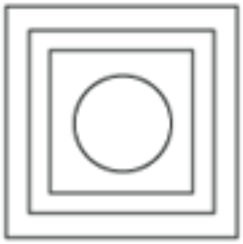
RESULT

```
GEOMETRIC ANALOGY
---------------------------------

A IS TO B
AS
C IS TO D1

(inside square (inside circle square)) IS TO (inside circle (inside square circle))
AS
(inside circle (inside square circle)) IS TO (inside square (inside circle square))
```

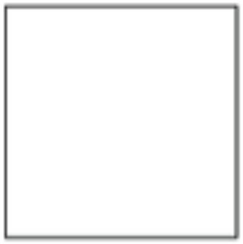
TEST CASE 4



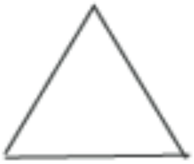
A



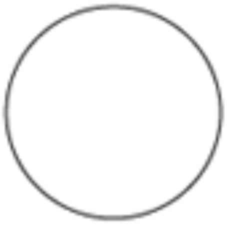
B



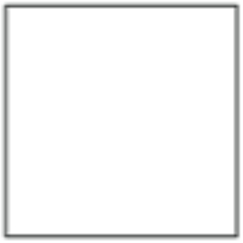
C



D1



D2



D3

From Sabine Bergler

REPRESENTATION

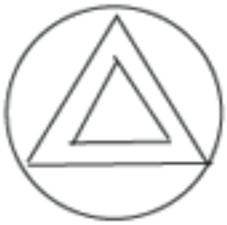
(define A '(inside square (inside square (inside square square))))
(define B '(inside square (inside square (inside square circle))))
(define C '(square))

(define D1 triangle)
(define D2 circle)
(define D3 square)

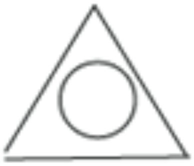
RESULT

This analogy cannot be solved with my analogy reasoner because the reasoner represents each segment of the problem as two shapes and a relation between the two shapes. In C, D1, D2 and D3 there is only one shape and thus no inter-shape relation. Further, my reasoner assumes that all the shapes in the 2nd segment of the "is to" relation are a subset of the shapes in the 1st segment. This is not the case in C is to D3. Finally my reasoner does not perform the reasoning required to solve this problem because it performs its reasoning by creating a pattern from A B and C and then inferring D. This sort of reasoning will not yield results for analogy problems of the sort in this test case.

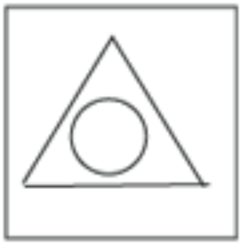
TEST CASE 5



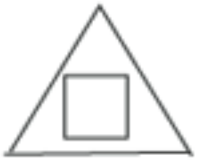
A



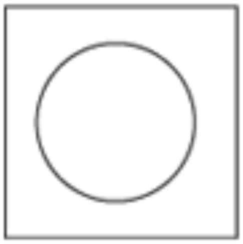
B



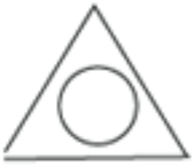
C



D1



D2



D3

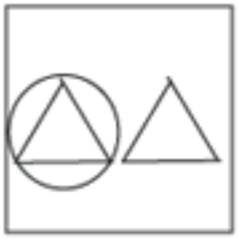
REPRESENTATION

```
(define A '(inside circle (inside triangle triangle)))
(define B '(inside triangle circle))
(define C '(inside square (inside triangle circle)))

(define D1 '(inside triangle square))
(define D2 '(inside square circle))
(define D3 '(inside triangle circle))
```

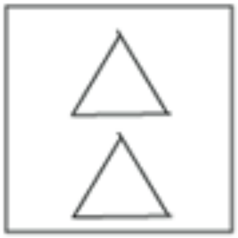
RESULT

```
GEOMETRIC ANALOGY
----------------------------------
A IS TO B
AS
C IS TO D1

(inside circle (inside triangle triangle)) IS TO (inside triangle circle)
AS
(inside square (inside triangle circle)) IS TO (inside triangle square)
```

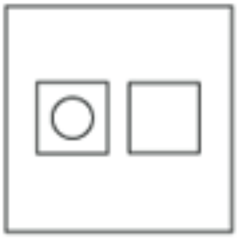
TEST CASE 6



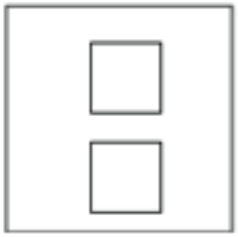
A



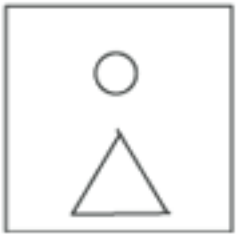
B



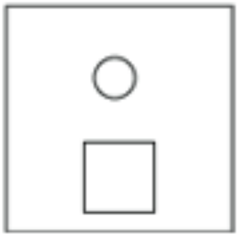
C



D1



D2



D3

This test case is interesting. Depending on how the segments are encoded the results vary. I will show three

TRIAL 1
REPRESENTATION

```
(define A '(inside square (right-of (inside circle triangle) triangle)))
(define B '(inside square (below triangle triangle)))
(define C '(inside square (right-of (inside square circle) square)))

(define D1 '(inside square (below square square)))
(define D2 '(inside square (below circle triangle)))
(define D3 '(inside square (below circle square)))
```

RESULTS

```
Sorry, could not solve this analogy
```

TRIAL 2
REPRESENTATION

```
(define A '(inside square (right-of (inside circle triangle1) triangle)))
(define B '(inside square (below triangle1 triangle)))
(define C '(inside square (right-of (inside square circle) square)))
```

```
(define D1 '(inside square (below square square)))
(define D2 '(inside square (below circle triangle)))
(define D3 '(inside square (below circle square)))
```

RESULT

```
GEOMETRIC ANALOGY
---------------------------------
A IS TO B
AS
C IS TO D3

(inside square (right-of (inside circle triangle1) triangle)) IS TO (inside square (below
triangle1 triangle))
AS
(inside square (right-of (inside square circle) square)) IS TO (inside square (below circle
square))
```

TRIAL 3

REPRESENTATION

```
(define A '(inside square (right-of (inside circle triangle1) triangle)))
(define B '(inside square (below triangle triangle)))
(define C '(inside square (right-of (inside square circle) square)))

(define D1 '(inside square (below square square)))
(define D2 '(inside square (below circle triangle)))
(define D3 '(inside square (below circle square)))
```

RESULT

```
GEOMETRIC ANALOGY
---------------------------------
A IS TO B
AS
C IS TO D1

(inside square (right-of (inside circle triangle1) triangle)) IS TO (inside square (below
triangle triangle))
AS
(inside square (right-of (inside square circle) square)) IS TO (inside square (below square
square))
```

The problem with this analogy is the two triangles in A and B. The knowledge representation and the heuristic used in the reasoner cannot solve this problem unless the two triangles are distinguished from each other. In trial 2 I represent the triangle as “triangle1” and “triangle” in both A and B. The results is D3. In trial 3 I only distinguish one triangle in A as “triangle1” and all other triangles are left as “triangle”. The result is D1. I designed this analogy with the intention of a result of D3, but after running reasoner for Trial 3 I see that the solution could just as easily be D1. This is interesting because it shows that one must consider how a geometric reasoner will resolve ambiguous analogies like this one.