# Project 3, CS7637

Shantanu Singh, shant0602@gatech.edu



## Project Approach

Following were the different stages of development of the AI agent implemented for RPM solution for Project 3.

### 1. Extension of Visual approach of Project 2

- I attempted to extend the visual approach as used in Project 2; however, I found the approach resulted in 6/12 correct Basic D and 4/12 correct Basic E problems.
- After investigation it was found that:
  1. For set D, the typical methods of binary transforms like union, intersection etc. and dark pixel difference were not enough.
  2. For set E, the binary transform methods were not working properly.

### 2. Modifications of visual approach

#### Phase 1

- For set D a new method had to be thought of and for set E, the binary transform methods were to be rectified by adjusting their threshold values for image matching.
- Analyzing the earlier 6/12 correct results in case of set D, it was found that problems like Basic D-02, D-03 achieved correct answer by union method due to low threshold of image matching, which was wrong.
- Such problems displayed an interesting feature: The total number of shapes along a row or column were same. In *Figure 1* ,along the first column there are total of 6 squares, 1 solid circle, 1 solid triangle and 1 plus. Same is the tally for all columns. This is used to predict the answer.
- In other words, the sum of no. of dark pixels along a row or a column is constant. The sum of percentage of dark pixels for all the matrices along row/column are compared with the other. If they are within 1% range of each other then this method is used to determine the answer.
- Including this method still resulted in 6/12 Basic D. However, as compared to previous case, where problems were getting solved by

Shantanu Singh, shant0602@gatech.edu


union due to threshold error, this time 1/12 was solved by intersection and rest 5/12 were by the 'sum of dark pixel method'.

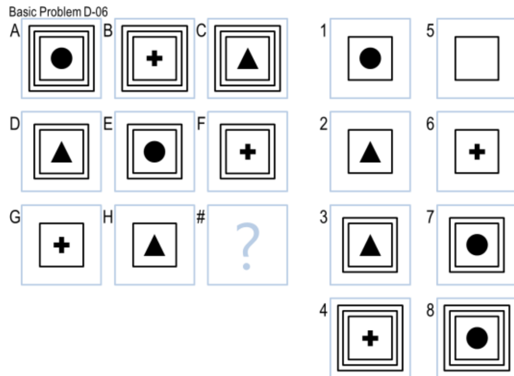



Figure 1 : Basic Problem D-06

## Phase 2

- In problems of the type *Figure 2* , it can be seen that total number of dark regions across a row/column are constant. For instance, along first row there are total of 6 black regions. This logic is used to predict the answer.
- However, using this logic there are more than one correct answer. For the problem in *Figure 2* this logic gives 2, 5, and 7 as correct answers.
- To resolve this, the answer choice which is not repeated in the question is finally chosen. Both 2 and 5 have already appeared in question as F and C, so that leaves out the option 7 as the correct answer.
- The way this algorithm works is that: we start scanning the image array row by row. The moment a dark pixel is encountered. Then its immediate four neighbors are checked for being a dark pixel. The location of dark neighbors is stored in a stack in form of tuples and those pixel values are set to a random number.
- This process is looped over with the start of each loop popping the last element from stack. By the time the stack clears it has gone through all the pixels of a continuous dark region.
- This approach solved four more problems in Basic and brought the count to 10/12 in Basic D.

Shantanu Singh, shant0602@gatech.edu



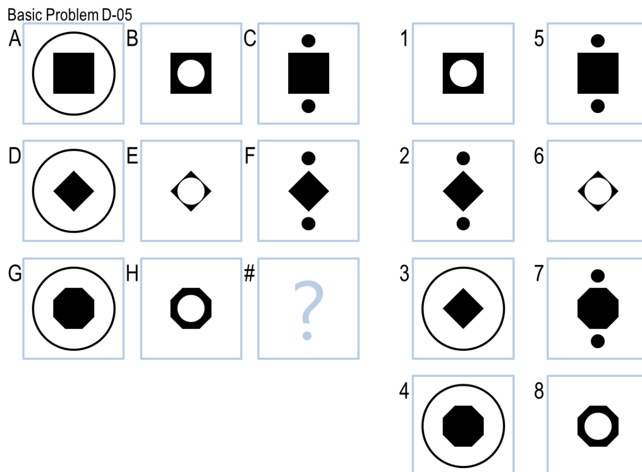



Figure 2: Basic Problem D-05

### Phase 3

- Raising the threshold of image matching to 98.5 % made the union method active and the number of correct answers went up to 6/12 for basic Problem E.
- From problem E-07 in *Figure 3*, the XOR operation resulted in an output as show in *Figure 4*. However, the answer is 4.
- For this purpose, a separate image matching threshold was created for XOR operation. The threshold was lowered to 96.
- After all the adjustments the tally for Basic problems E came up to be 9/12.

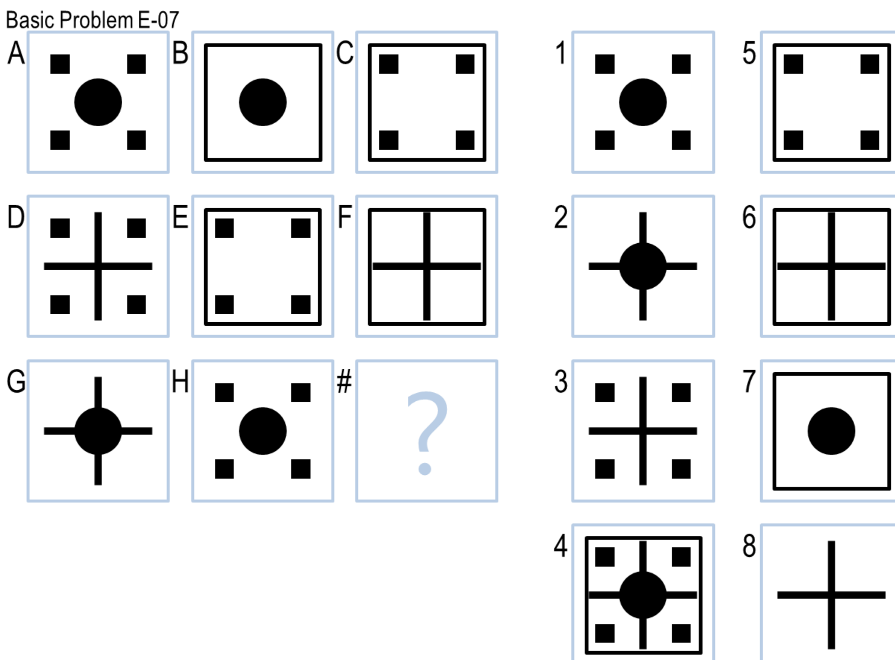



Figure 3 : Basic Problem E-07

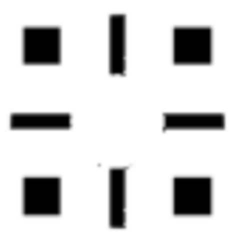


Figure 4: Expected result of Basic Problem E-07 by XOR operation

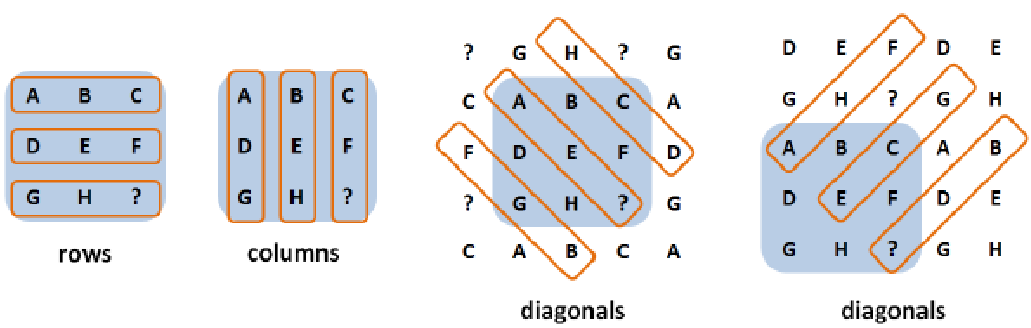



**Figure 5**: Collinear triplets and parallel set of collinear triplets in 3x3 matrix *[1]*

Shantanu Singh, shant0602@gatech.edu




| | Rows | Columns | Diagonals | |
|---|---|---|---|---|
| Pairs | A : B :: H : ?<br>B : C :: H : ?<br>D : E :: H : ?<br>E : F :: H : ?<br>G : H :: H : ?<br>A : C :: G : ?<br>D : F :: G : ? | A : D :: F : ?<br>D : G :: F : ?<br>B : E :: F : ?<br>E : H :: F : ?<br>C : F :: F : ?<br>A : G :: C : ?<br>B : H :: C : ? | F : G :: E : ?<br>G : B :: E : ?<br>H : C :: E : ?<br>C : D :: E : ?<br>A : E :: E : ?<br>F : B :: A : ?<br>H : D :: A : ? | F : H :: D : ?<br>H : A :: D : ?<br>G : C :: D : ?<br>C : E :: D : ?<br>B : D :: D : ?<br>F : A :: B : ?<br>G : E :: B : ? |
| Triplets | A : B : C :: G : H : ?<br>D : E : F :: G : H : ? | A : D : G :: C : F : ?<br>B : E : H :: C : F : ? | F : G : B :: A : E : ?<br>H : C : D :: A : E : ? | F : H : A :: B : D : ?<br>G : C : E :: B : D : ? |

**Table 1:** List of combinations in a 3x3 matrix *[1]*

| **Transformation No.** | **Unary (Applied on A and the result compared with last element of doublet)** | **Binary (Applied on first 2 elements of triplet and the result is compared with last element of triplet)** |
|---|---|---|
| 1. | Identity | Union |
| 2. | Rotate90 | Intersection |
| 3. | Rotate180 | A-B |
| 4. | Rotate270 | B-A |
| 5. | Identity-flip | XOR |
| 6. | Rotate90-flip | |
| 7. | Rotate180-flip | |
| 8. | Rotate270-flip | |

**Table 2:** Unary and Binary transformations

# Final Agent Development

## 1. Assumptions for the Current Model Representations in Agent

- The representation of images are 2-D arrays of grayscale pixels. Thus, each pixel is associated with a single intensity value (0: white and 1: black). The code explicitly converts the supplied image to grayscale.
- While matching two images the agent displaces the image by 1 pixel in all directions to account for error tolerance in the position of pixels in image.

## 2. Working of the final agent

The agent implements **visual approach** to solve the RPM. Images are loaded from the dictionary of Raven figure objects.

Pseudocode of Agent's algorithm

STEP 1: Load each problem

STEP 2: Apply the following transformation in given order:

STEP 3: If transformation for two row triplets is UNION:

Generate the missing entry and get the correct answer by testing against each possible choice, GOTO STEP 11.

Repeat STEP 3 for two column triplets

STEP 4: If transformation for two row triplets is INTERSECTION:

Generate the missing entry and get the correct answer by testing against each possible choice, GOTO STEP 11

Repeat STEP 4 for two column triplets

STEP 5: If transformation for two row triplets is SUBTRACTION:

Generate the missing entry and get the correct answer by testing against each possible choice, GOTO STEP 11

Repeat STEP 5 for two column triplets

Shantanu Singh, shant0602@gatech.edu


STEP 6: If transformation for two row triplets is XOR:

Generate the missing entry and get the correct answer by testing against each possible choice, GOTO STEP 11

Repeat STEP 6 for two column triplets

STEP 7: If transformation for two row triplets is by REGION COUNT PROGRESSION:

Get the correct answer by testing the pixel difference against each possible choice,
GOTO STEP 11

Repeat STEP 7 for two column triplets

STEP 8: If transformation for two row triplets is by DARK PIXEL SUM:

Get the correct answer by testing the pixel difference against each possible choice,
GOTO STEP 11

Repeat STEP 8 for two column triplets

STEP 9: If transformation for two row triplets is by DARK PIXEL RATIO:

Get the correct answer by testing the pixel difference against each possible choice,
GOTO STEP 11

Repeat STEP 9 for two column triplets

STEP 10: If no transformation matches return -1.

STEP 11: Repeat from STEP1, until all problems are loaded.

- Both the triplets of row or column as shown in **Table 1** are checked for the transformation patterns to ascertain strong relationship.
- The dark pixel percentage difference threshold is taken to be 1%

Shantanu Singh, shant0602@gatech.edu


# Performance and efficiency

## 1. Performance of the Agent

| | **Results of the final version of Agent** |
|---|---|
| Basic D | 10/12 |
| Test D | 8/12 |
| Raven D | 6/12 |
| Challenge D | 2/12 |
| Basic E | 9/12 |
| Test E | 8/12 |
| Raven E | 8/12 |
| Challenge E | 3/12 |

**Table 3:** Performance of the agent

As seen from the table above the performance of Agent is decent for Basic and Test problems D and E, however performance is poor in challenge problems.

## 2. Efficiency of Agent

| | Time (seconds) |
|---|---|
| Basic Problem D | 46 |
| Basic Problem E | 33 |
| Project 3 | 916 |

Table 4: Efficiency of Agent

For project 3, the agent takes almost **15 minutes** to solve all problems. From the performance it can be said that agent is **general** enough as it performs decent on test, basic and Ravens problem. The **performance on test and basic** are almost same.

Shantanu Singh, shant0602@gatech.edu



# Limitations

The **overall limitations** of the agent are:

**Type 1**: The agent can't recognize shapes or edges. Thus, it fails on many problems in challenge E.

**Type 2**: For 3X3 problems the rotation transformations are not incorporated in the agent due to lack of time.

**Type 3**: The agent still can't deal problems with moving shapes as shown in *Table 5*.

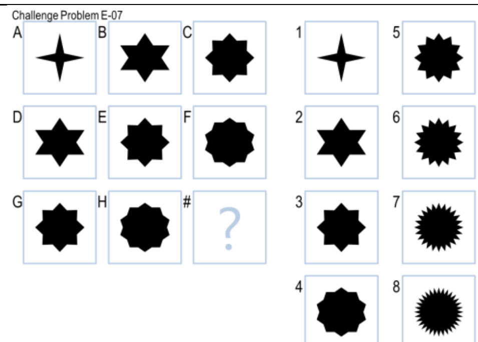
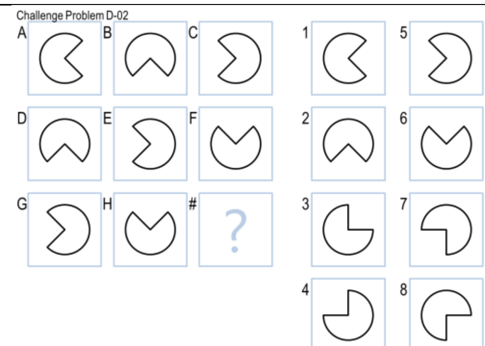
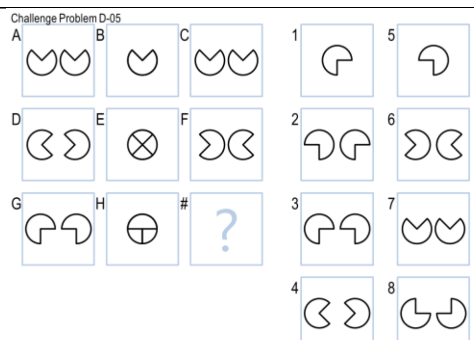
| Type 1 | Type 2 |
|---|---|
|   |   |
| Type 3 | |
|   | |

**Table 5:** Types of Cases where the agent failed

## Connection with Human Cognition

- Although the agent in consideration is essentially based upon a set of conditions written in form of an object-oriented program; still it has a weak connection with the process of human cognition. The agent is basically a semantic network. Different states are generated and tested for success to conclude the best path for the mapping.
- The agent has been revised multiple times by learning from its mistakes. It has been improved by incorporating new methods to tackle different type of RPM problems. This process is identical to the way humans learn from experience.
- Up to a brief extent, the problem-solving procedure of agent reflects the human problem-solving behavior in the sense that the agent matches the images step by step based on a sequence of various transformations. However, unlike humans the agent cannot cut short a mapping by recognizing that this path is leading to a wrong solution. The agent would go till the end and match the transformation with a similarity measure unlike a human who can abandon a path just by looking and predicting that path is not going to match.

Currently the agent is not really a true AI as its not learning from its past experiences to predict new transformations and relationship. Thus, it doesn't resemble the cognition of a human solving RPM in this aspect. However, the pattern of problem solving does resembles the approach of humans. Thus, it would be apt to say that the methodology used to solve the problem partly resembles to that of humans.

Shantanu Singh, shant0602@gatech.edu



# References

[1] M. Kunda, "VISUAL PROBLEM SOLVING IN AUTISM, PSYCHOMETRICS, AND AI: THECASE OF THE RAVEN'S PROGRESSIVE MATRICES INTELLIGENCE TEST," 2013.