CS7637: Knowledge-Based AI: RPM Milestone 1

Jing Gan
jgan34@gatech.edu

Abstract—First, I discuss my approaches to four problems on Raven's test. Second, I describe the details of how my agent will be designed. Last, a discussion on the biggest challenges in designing the agent is provided.

1 FOUR PROBLEMS AND MY APPROACHES

1.1 Problem 1 - Basic Problem B-03

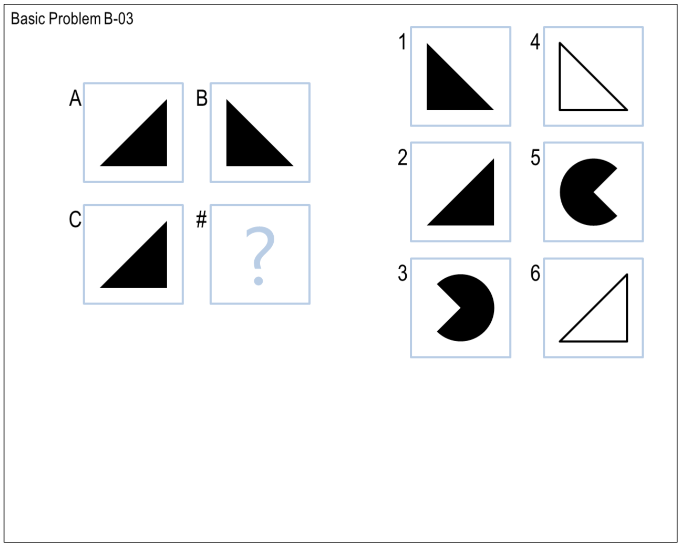

Figure 1— Basic Problem B-03, obtained from the Basic Problems B folder.

I can come up with two approaches:

1) **Reflection**. B can be viewed as a reflection of A across a vertical axis. The vertical axis locates between A and B. If I think of a similar vertical axis between C and the graph we need to determine, I will get Choice 1 which will be the reflection of C in this case.
2) **Rotation**. B can be viewed as A rotating clockwise for 90 degrees or anti-clockwise for 180 degrees. In either case, if I rotate C similarly, I still get Choice 1 as my answer.

1.2 Problem 2 - Basic Problem D-08

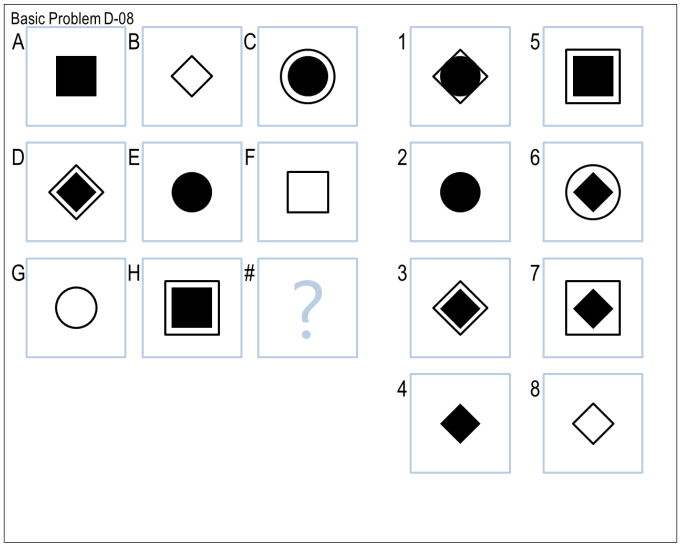

Figure 2— Basic Problem D-08, obtained from the Basic Problems D folder.

I can think of two approaches:

1) First, I notice that each of the first two rows contains three distinct shapes: square, diamond, and circle. Therefore, I would expect the answer in the shape of a diamond. Second, in each row, the shape is filled in three different ways: 1) filled with black, 2) filled with nothing, and 3) filled with black but in the same smaller shape. If I follow this pattern in the third row, I should expect a diamond filled with black. Choice 4 meets this requirement perfectly.
2) Horizontally, each row except for the third row has a diamond. Vertically, each column except for the third column also has a diamond. Diagonally, shapes are filled with black in the same way. Therefore, the answer is also Choice 4, which is a full black diamond.

1.3 Problem 3 - Challenge Problem B-05

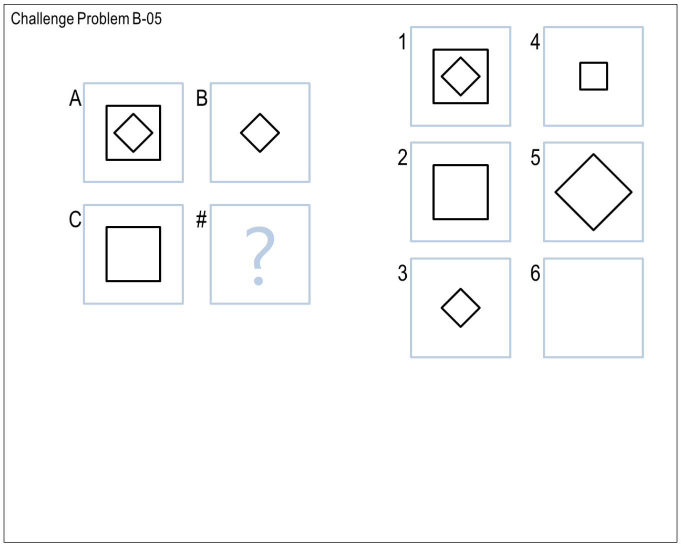

Figure 3— Challenge Problem B-05, obtained from the Challenge Problems B folder.

There are two approaches in my opinion:

1) B can be viewed as the result of A after removing the outside shape. So, the inner diamond is preserved. Similarly, if I remove the outside shape of C, I get nothing left. The answer is Choice 6.
2) To get B, the outside shape of A is rotated clockwise 45 degrees, so we get two parallelly aligned diamonds. Then, the outside shape shrinks to the same size as the inner one. Following this logic, if I rotate C first, and shrink its size, I will get a diamond just like Choice 3.

2 HOW DO I DESIGN THE AGENT?

2.1 Use verbal representation

I will use a verbal representation or in other words, knowledge representation using semantic networks when designing my agent. First, I would like to represent all the objects in the inputs. Second, I will add labeled links (above, inside, top-right, etc.) among these objects to represent the relationships. Third, I will examine the two inputs (in a $2 \times n$ Raven's test) or three inputs (in a $3 \times n$ Raven's test) together to add transformation information using labeled links (deleted, unchanged, expanded, etc.) as well.

After constructing this semantic net which records the given knowledge, I can ask my agent to apply the same representation on the last row to generate the expected answer. If it can be found in the choices, the agent could conclude that the given Raven's test is solved. Or, I can tell the agent to build representations in the last row after plugging in all possible choices and choose the one that is closest to the previous representations as the final answer.

2.2 Use heuristic methods

I would also like to use heuristic methods in my agent. Just like what I discuss in Problem 1 - Basic Problem B-03 and Problem 2 - Basic Problem D-08, the agent can record whether the number of certain shapes has changed vertically and horizontally, the distance of shifts, the degrees that objects rotate, etc. It can also use the calculation of the darkened pixels or the ratio of the darkened areas in Problem 2 - Basic Problem D-08 to make decisions.

2.3 Decision among strategies

My agent should be able to decide on multiple strategies by itself. As I presented earlier, several approaches to the same test are usually used. Ideally, the agent will first generate answers via different methods and then compares the results to each other and the given choices. If the same result is obtained, the test is solved. Otherwise, the agent needs to decide which method to follow.

It is worthwhile to mention that even using the same approach, like in Problem 1, the agent could face choices between reflection and rotation if different results are obtained. In this case, it is helpful to choose the match by weight, i.e., assign points to transformations and decide by scores.

3 BIGGEST CHALLENGES IN DESIGNING THE AGENT

The first challenge could be the lack of vocabulary in building a semantic network. If I fail to program enough keywords to represent the relationships among objects, it is no doubt that my agent will fail to capture enough knowledge from the input and thus detect the correct answer effectively and correctly.

The second challenge is that when facing a very large semantic network including many nodes and complex relationships, how could my agent rank them? In other words, how does the agent know which relationship is of the greatest importance and which is not?

The third challenge is how can I improve the agent's performance efficiently. Should I adjust it case by case, or can I find a more productive way to improve its overall “intelligence”?

I believe I will gain more insights as I continue the study of this course.