CS7637: Knowledge-Based AI: RPM Project: Milestone 2

Jing Gan
jgan34@gatech.edu

1 QUESTION 1

How does your agent currently function?

My agent first considers reflections or flipping of Figure A to Figure B or to Figure C. After loading Figure A, it flips it either according to the X-axis or the Y-axis. Then, it utilizes the PIL.ImageChops.difference() function to check the differences between Figure A and Figure B, and Figure A and Figure C. If an exact match is determined, the agent can tell whether the question should be solved vertically or horizontally. According to this result, it compares the 6 choices to either Figure B or Figure C.

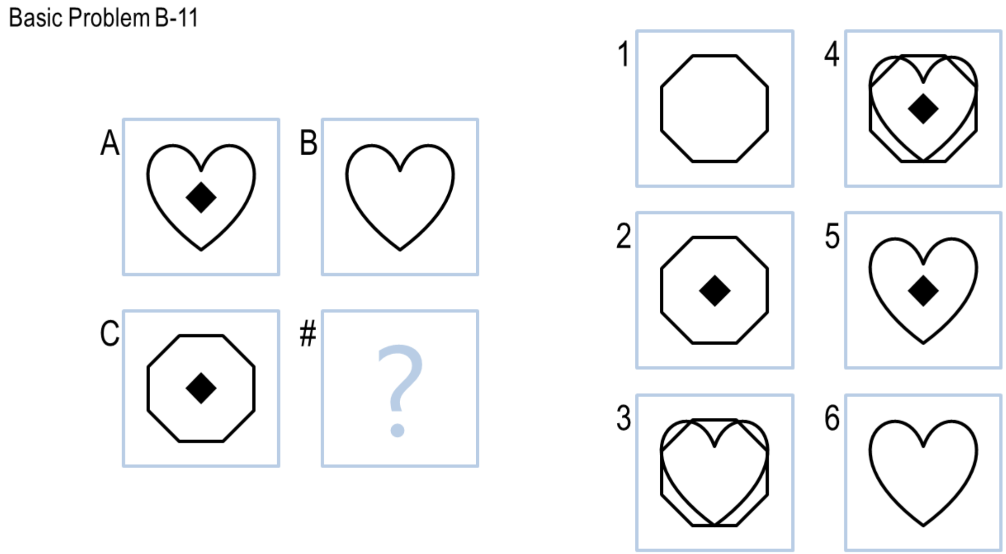
If reflection transformation fails, it consider a rotation of 45°, 90°, 135°, 180°, 225°, 270°, and 315°. Again, it checks A to B first and then A to C. After applying such rotations to Figure A, if Figure B is an exact match, it finds the solution.

2 QUESTION 2

How well does your agent currently perform? How many problems does it get right on the Set B problems?

My agent performs just okay in Set B problems. More specifically, when tested locally, my agent can solve Basic Problem B-1 to B-8, and it fails to provide correct answers for Basic Problem B-9 to B-12.

When I investigate these four questions closely, I realize that they all either require more transformations or other methods of comparing the results and given figures. For example, in Basic Problem B-11, shown in Figure 1, simply conducting a rotation or a reflection transformation does not make Figure A become Figure B or C. Instead, it requires either the agent to detect the frame that stays the same or to make a deduction between the two graphs so that the outside shape disappears and only the filled diamond shape remains. Unfortunately, my agent's algorithms are not adequate to perform either task.



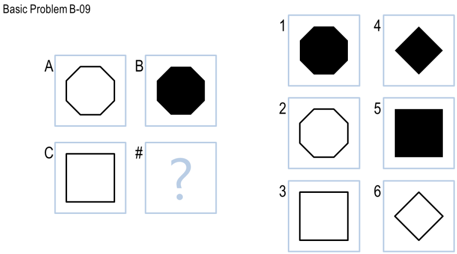
3 QUESTION 3

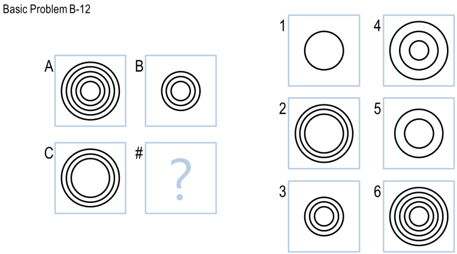
What problems does your agent perform well on? What problems (if any) does it struggle on? Why does it struggle?

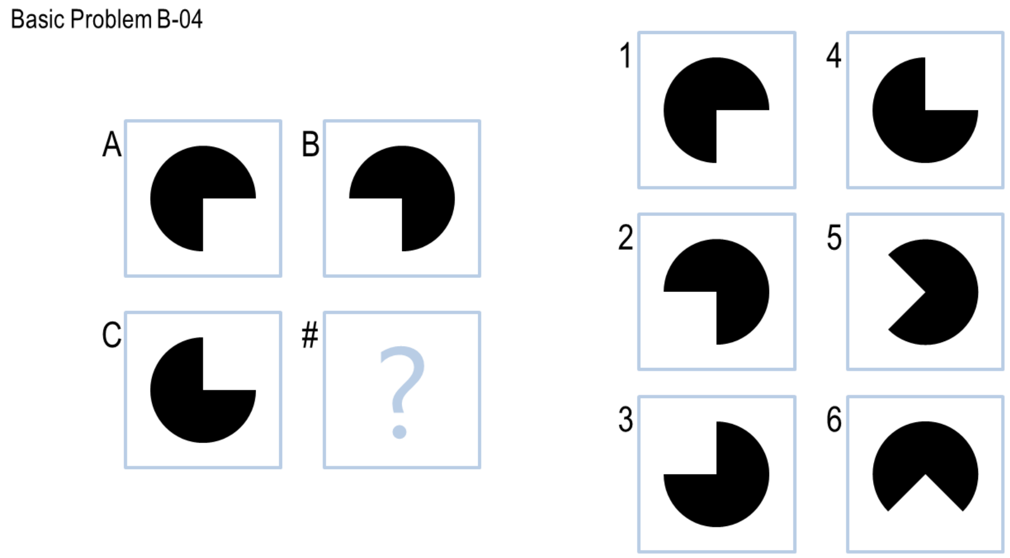
My agent performs well on questions involving simple **rotations or reflections** such as question Basic Problem B-04 shown in Figure 2. I have included the common rotation degrees in the algorithm (45°, 90°, 135°, 180°, 225°, 270°, 315°)

to make sure the agent covers enough possibilities. After detecting such transformation from Figure A to B or Figure A to C, it compares Figure B or C to the choices pixel by pixel to find an exact match and return the answer.

As mentioned in Question 2, it struggles with questions involving frames which cannot be achieved by rotating or flipping Figure A. More specifically, when Figure B or C is created by filling the frame in Figure A like Basic Problem B-09, or deleting inside/outside frames like Basic Problem B-12, my agent fails. The reason is that I fail to come up with a suitable algorithm to detect and quantify such transformation.







4 QUESTION 4

How efficient is your agent? Does it take a long time to run? Does it slow down significantly on certain kinds of problems?

My agent is very efficient in terms of solving questions quickly. When tested locally, it is able to provide answers in seconds. And when uploaded to Gradescope and faced with more questions of more transformations, it significantly slows down. Although I have no access to the content of questions on Gradescope, I can expect that, as discussed earlier, it struggles with questions involving frames which cannot be achieved by rotating or flipping. Also, there may be other problems that involve even more complex transformations than deleting or filling the frames that my agent can't solve.

5 QUESTION 5

How do you plan to improve your agent's performance on these problems before the final project submission?

My plan is straightforward. I would like to focus on frame problems specifically and try to deal with the filling and deleting transformations. I believe I need to learn some new libraries of Python which are more capable of performing such tasks.

I would also like to manually go through other Basic and Challenge problems and figure out what other transformations I need to pay attention to and improve the agent's algorithm accordingly.

6 QUESTION 6

How do you plan to generalize your agent's design to cover 3x3 problems instead of just 2x2 problems?

This question is challenging. At this stage, I would probably use Means-Ends analysis to generalize my agent's design. As we know, in a 3 by 3 problem, we usually have Figure A transform to Figure B first, and then to Figure C. The difference (end) could persist or upgrade or downgrade during this process. Therefore, my agent should be able to detect, quantify, and analyze such step-by-step transformations and apply similar transformations to the third-row graphs and find a suitable answer.

7 QUESTION 7

What feedback would you hope to get from classmates about how your agent could do better? What challenges do you think could benefit from someone else's feedback?

In fact, any suggestions and feedback are greatly welcomed. If you have more ideas on dealing with frames problems, please let me know!