

Amol Parikh
amolparikh21@gatech.edu
November 2, 2014

Project 3

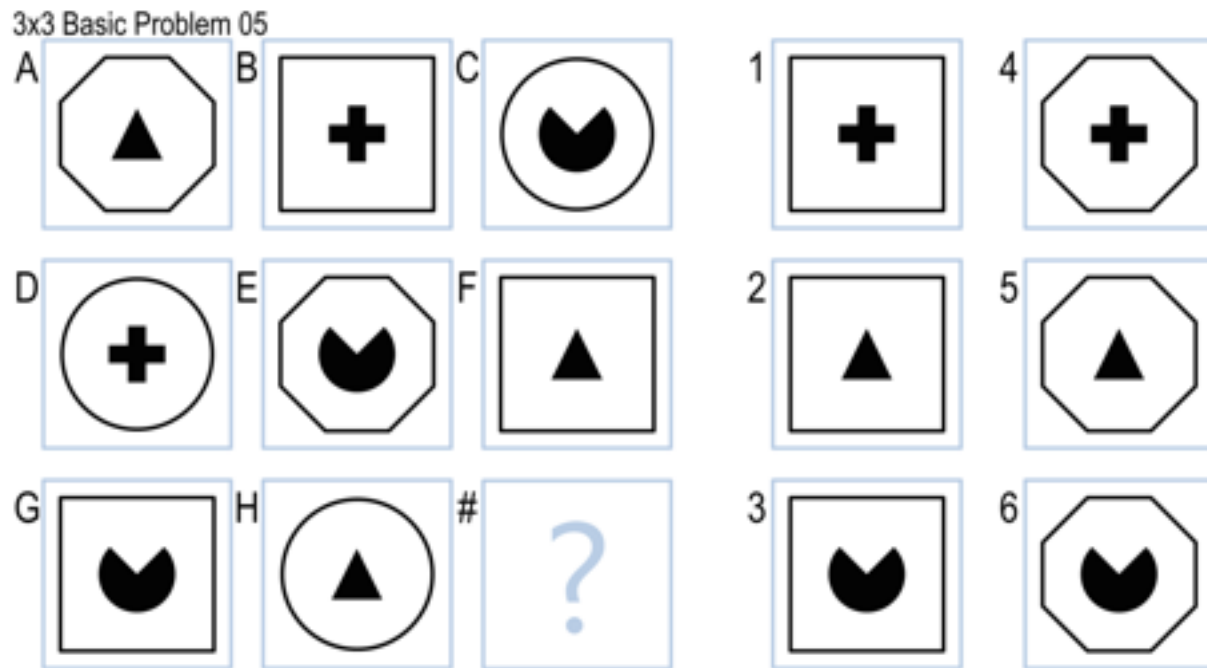
Reasoning the Agent Uses.

The agent that I have created for this project uses a combination of Cased Based Reasoning and Generate and Test. The agent initially goes through all the transitions from one figure to another in the file and stores all of them in a cvs file. So it reads through all transitions A -> B, B -> C, D -> E, E -> F, G -> H and stores the entire semantic network of each of these transitions in a cvs file. It then reads Figure H and for the shape of the object in figure H, it searches for the correct transition of that figure in the csv file and applies that transition to the figure along with all the transitions of the attributes of that figure in the cvs file. It does so by reading through all the rows in the csv file that are after the figure that was found earlier till it encounters an attribute shape again which would indicate the end of the list of attributes of that figure. It then reads the next object of figure H and repeats the procedure. In this way it generates every object in figure I. Then comparing all the generated objects in figure I with the attributes of the option provided it chooses the correct option.

In certain cases this approach does not work because the figures aren't related by just the transition but also by the difference in the number of objects, angles, sides etc. For this the agent considers a special case for angles and for the number of figures. It calculates the difference in the number of objects in figures between A and D, B and E, C and F, D and G, E and H. It then takes figure F and then taking into consideration the difference determines the number of figures in I. It then adds all attributes using the previous approach and comes to the correct option. In case of angles as well, it calculates the differences in the angles in consecutive figures and then applies the same difference to generate the object in figure I.

How it comes to some of the correct answers?

Taking an example of questions already given I will explain how my agent comes to the correct answers.



In the above problem the agent stores the following transitions in a csv file.

The csv file looks something like this :

Shape, Octagon , Square

Shape, Triangle, Plus

Fill , Yes, Yes

Shape, Square , Circle

Shape, Plus, Pac-man

Fill , Yes, Yes

Shape, Circle, Octagon

Shape, Plus, Pac-man

Fill , Yes, Yes

Shape, Octagon , Square

Shape, Pac-man, Triangle

Fill, Yes, Yes

Shape, Square , Circle

Shape, Pac-man, Triangle

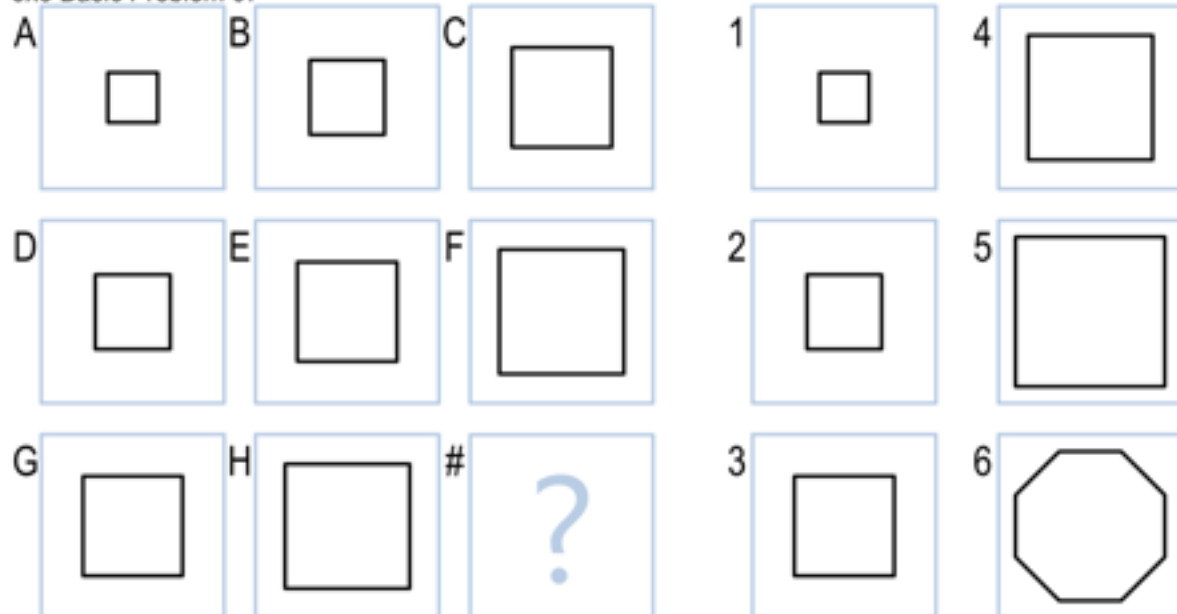
Now it reads each object in the figure H and then searches for the shape in the csv file.

It gets the shape circle and searches for circle in the 2nd column in the csv file. After the search it stores the 3rd column in an array list as the generated Figure I. So it generates Octagon. It then take the second object Triangle and searches the 2nd column for triangle. It stores the corresponding 3rd column (Plus) in the array list of figure I. It then continues storing the attributes of triangle which is fill. It similarly stores the fill attribute as Yes in the array list. It continues doing so till it reads the next row as shape which indicates the end of the attributes of that figure.

Why the agent makes some of the mistakes it does.

One of the major drawbacks of the approach that i have used is that i case the figure that the agent encounters is completely new in terms of its shape, attributes then it cannot correctly generate the correct answer. So in case of the problem below:

3x3 Basic Problem 07

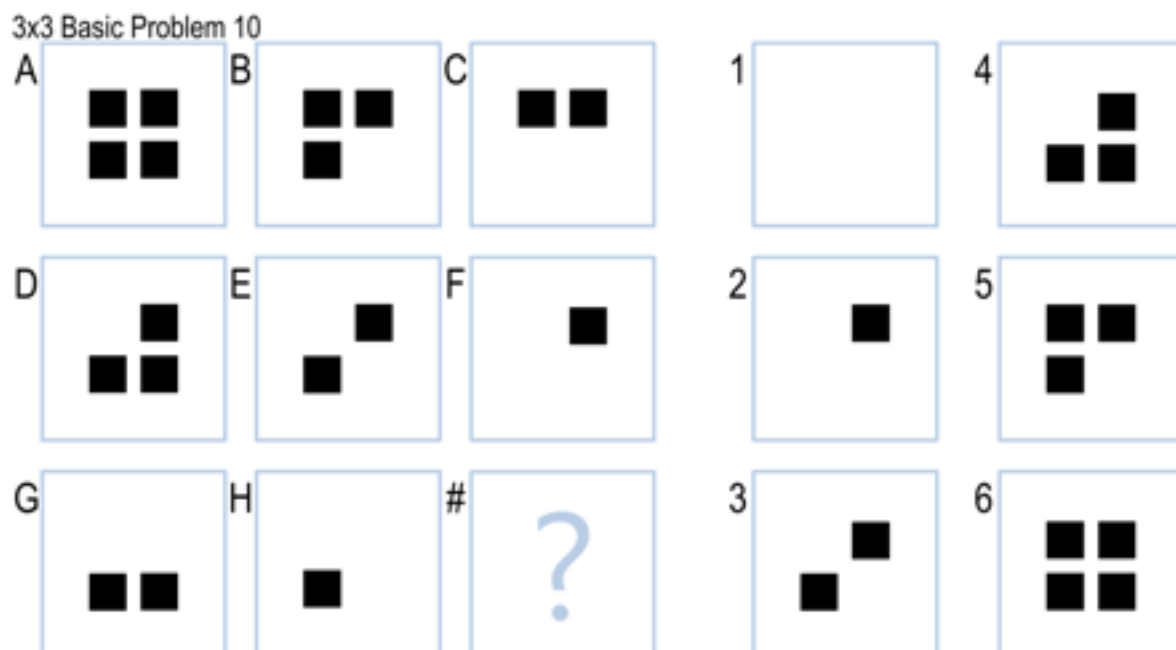


The csv file has a number of rows with the shape square and its attributes for each figure. So the different shapes that it has stored are : Very Small, Small, Medium and Large. Now it reads the object in figure H and sees that the shape is square and the size is large. It searches for large in the 2nd column and it does not find it. this is because it has never encountered a transition where the shape size goes from large to something else. In the transition from E to F the size goes from medium to large but that does not help the agent to determine the transition from large to the correct answer(i.e) very large.

This is one of the drawbacks of the agent that i have determined.

What could be done to improve the agent?

The main thing that could be done to improve the agent is that it should also take into account the transitions in the columns. So if it stores all the transitions from A to D, D to G, B to E, E to H, C to F. This would improve the agent in questions such as :



In this case the combinations of the transitions between the rows and the columns would result in the correct answer.

Another thing that could also make the agent more efficient is to make it learn that if the transitions are in increasing from very-small to small to medium to large then the next transition should be to very large. In this way it will be able to solve many more problems.

How long does it take to run?

The agent takes a second to run and give the final answers. However, the fact that we are writing to a csv file takes a little longer than doing it just in memory as the time taken to read and write to a disk is longer.

Problems that have a long list of attributes would take a little longer as more number of rows have to be written in the csv file and also a longer search will take place when searching for attributes in the file. This is a little inefficient as compared to searching when loaded onto memory.

Describe the relationship between your agent and human cognition.

In case of humans , when we solve problems such as ravens matrices, we also look for cases where we have encountered a similar case and then generate the final answer. So the agent also does the same thing. It looks for cases which it has already encountered a similar problem and then uses the outcome of those cases to generate an answer. This is similar to what humans do in most cases. However, humans learn on the way and so in certain cases when extrapolation is required, humans can easily come up with an answer where as the agent cannot extrapolate to get the correct answer. Another difference between how humans solve and the agent solves is that when there is direct relation between the first and the final answer, the agent still tries to search through the entire csv file to get to the answer.