RPM PROJECT: MILESTONE 1 JOURNAL

David Adamashvili
dadamashvili3@gatech.edu

1 HUMAN APPROACH TO RAVEN'S PROGRESSIVE MATRICES

The way humans approach visual problems such as Raven's Progressive Matrices (henceforth abbreviated as **RPM**) might at first seem arbitrary and non-algorithmic, different for every specific matrix. However, the field of Knowledge Based AI (**KBAI**) has shed light on our abstract thinking processes by analyzing how various knowledge representations can be embedded onto computing systems. More general techniques such as **Semantic Networks** and **Means Ends Analysis** have been developed that can work for various different problems. However, it is clear that if humans are using a similar approach to this, they are doing it subconsciously, as they are extremely efficient compared to computers at finding visual analogies and noticing patterns. Also, from personal observations, humans tend to think in more abstract terms, and choose between visual transformations, verbal analogies and heuristics depending on the problem. Let us look at a few examples of how humans would go about solving RPM problems

1.1 Example 1

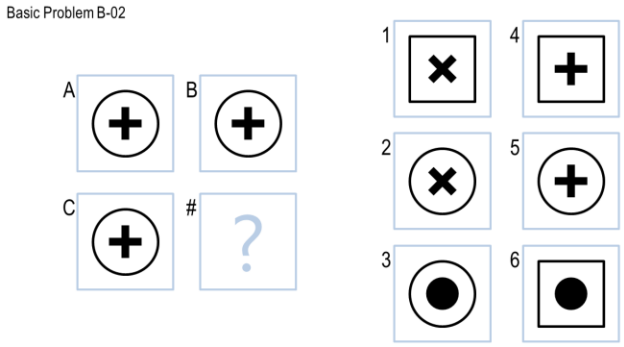

Figure 1 - Example of heuristic reasoning

For humans, the solution to the problem in Figure 1 is obvious. We instantly recognize that the three shapes A, B and C are the same. Therefore, continuing the pattern, D must also be the same. Therefore, 5 is the correct answer.

1.2 Example 2

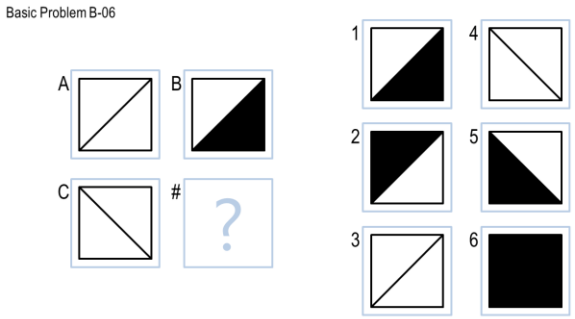

Figure 2 - Example of visual transformations

Here, we can look at the problem in two ways. Going top to bottom, we see that we go from A to C by flipping (reflecting) it about the vertical axis. By means of analogy, D should be the result of flipping B, meaning the correct answer is 5.

But, we can also look at the pictures as two triangles glued to each other. Going left to right, we see that we go from A to B by filling in the bottom triangle. The answer comes out to be 5 again. This may not be the case in other similar problems! There can actually be more than one correct answer.

1.3 Example 3

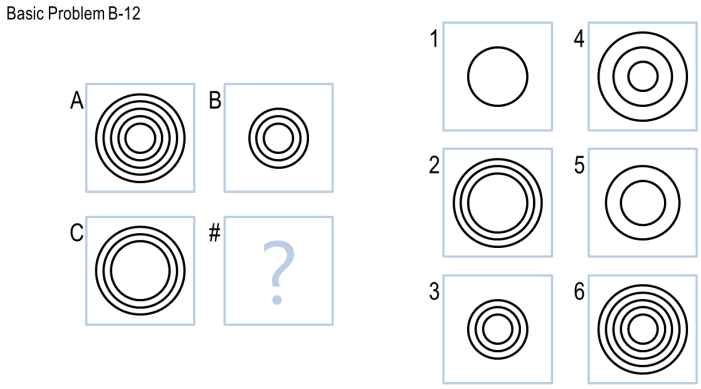

Figure 3 - Relative positions and counting

In Figure 3, going from A to B, the outer two circles disappear. From A to C, the inner two disappear. Again, by analogy, D should be whatever we get by removing the inner two from B or the outer two from C. In both cases, the answer will be 1.

1.4 Example 4

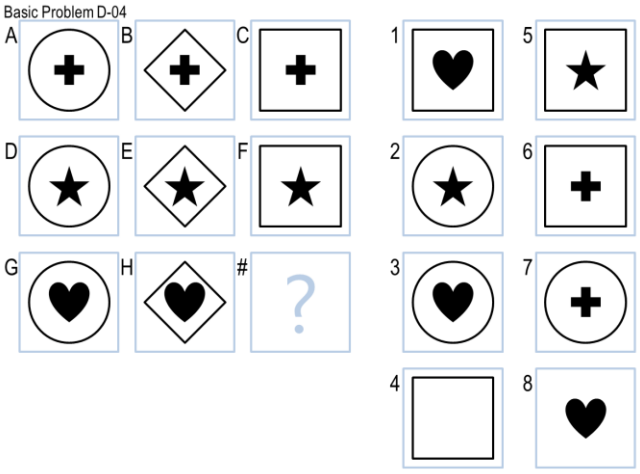

Figure 4 - 3x3 matrix example

Here, a human would notice that there are two shapes in all cells, one inside another. The outer shapes are all same in every column and the inner shapes are all same in every row. Hence, the answer must be 1.

2 DESIGNING AN AGENT TO SOLVE RPM PROBLEMS

The simplest way to design an agent for finding patterns in RPM matrices would be by using heuristic methods on the pictures like counting black pixels and applying geometric and logical transformations. In most simple cases, creating verbal representations and semantic networks would be unnecessary.

For example, the problem in **Figure 1** could be solved by counting the black pixels in every picture and noticing that this number does not change when going right (A to B) or down (A to C). Solving the same problem using semantic networks would be a lot harder, as one would need to classify all of the shapes, some of which are not trivial.

The same strategy would not work for the problem in **Figure 2**, as there are the same number of black pixels in 1, 2 and 3. Here, we would also need to apply geometric transformations such as flipping the image and then checking for image equality. In general, other geometric transformations such as rotations should also be tried.

The problem in **Figure 3** might seem tricky at first, but notice that going from left to right decreases the number of pixels by some amount **X**, and going from top to bottom decreases them by **Y**. Checking all the answers will show that 1 fits this pattern. The same is true for **Figure 4**, the pairwise pixel differences are constant throughout rows and columns.

We should also consider operating on the pictures using logical operators. For example, if we can get the cell C by using the logical **XOR** operator on A and B.

If heuristic methods are exhausted and fail, we can lean back on creating semantic networks from verbal representations of the cells. This will be needed in harder problems, where the shapes in the pictures change relative positions, while everything else stays the same. I will implement this at a later date if necessary, as for now, heuristic methods seem to be enough.

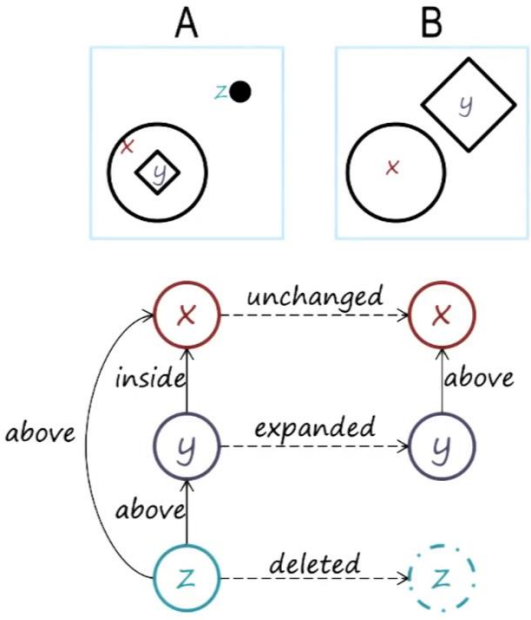

Figure 5 - Example of a semantic network

3 CHALLENGES WITH DESIGNING AN AGENT

Creating heuristic algorithms based on pixel counting and image transformations for finding patterns should not be too difficult. However, if we for example want to create a heuristic such as "counting the number of pentagons in each cell", then the problem becomes a lot more challenging.

Luckily, the open source computer vision library OpenCV helps us deal with this. But, even with the help of the library, implementing such heuristics is not straightforward. It contains a similarity metric function for the shapes, a way to find the centroid of a figure and also a way to calculate the principal axis of any given shape. The latter will be used to find the angle of rotation between two similar figures. All of these functionalities will need to be properly studied and tested.

Lastly, we would also need to find out if one figure is a scaled up version of another. For this, a function for scaling shapes perpendicular to its contours would be needed. I have not found any prebuilt functions inside OpenCV that can do this out of the box, so we might need to implement it ourselves.

After all of this is done, if heuristic algorithms are not enough, semantic networks will need to be developed. Every possible object mapping between cells will need to be tried, and every answer ranked.

All of the abovementioned steps are ordered in increasing difficulty, but hopefully heuristic algorithms will be enough to solve most of the basic problems in the set.