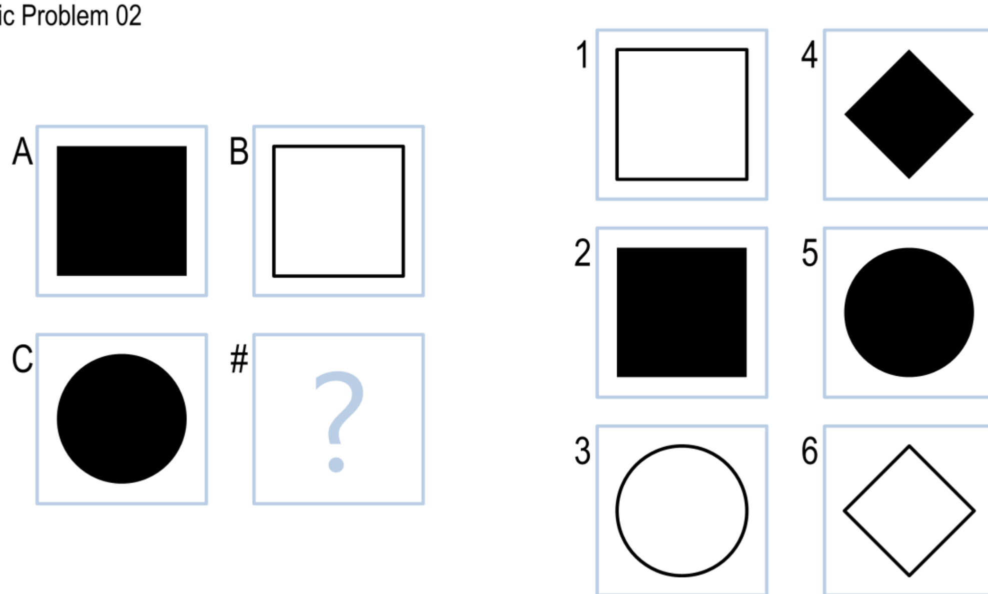
Brad Ware
bware8@gatech.edu
Assignment 5

Using Analogical Reasoning to solve Raven's progressive matrices

When I actually analyzed how I solve problems that arise throughout the day, it amazed me how often I referred to a situation that was similar to the present. I can instantly face a problem and retrieve a situation in the past that was similar, applying the same analysis to draw a conclusion. The human brain is fascinating and extremely powerful, as we subconsciously retrieve situations from the past and apply them to the present. If humans naturally try to solve problems in this fashion, then it must be a strong strategy for finding a solution. Analogical reasoning is the process of categorizing situations as they are experienced and storing them for future scenarios. Then to solve a problem encountered in the future, one analogically maps to a past situation in memory (retrieval) and applies the same strategies to hopefully reach a conclusion for the current problems. For this strategy obviously one becomes more likely to solve problems as they encounter more, as they have a bigger memory bank for retrieval. As new problems or situations are encountered one typically recognizes this because it doesn't map to anything, and will store the outcome for later use.

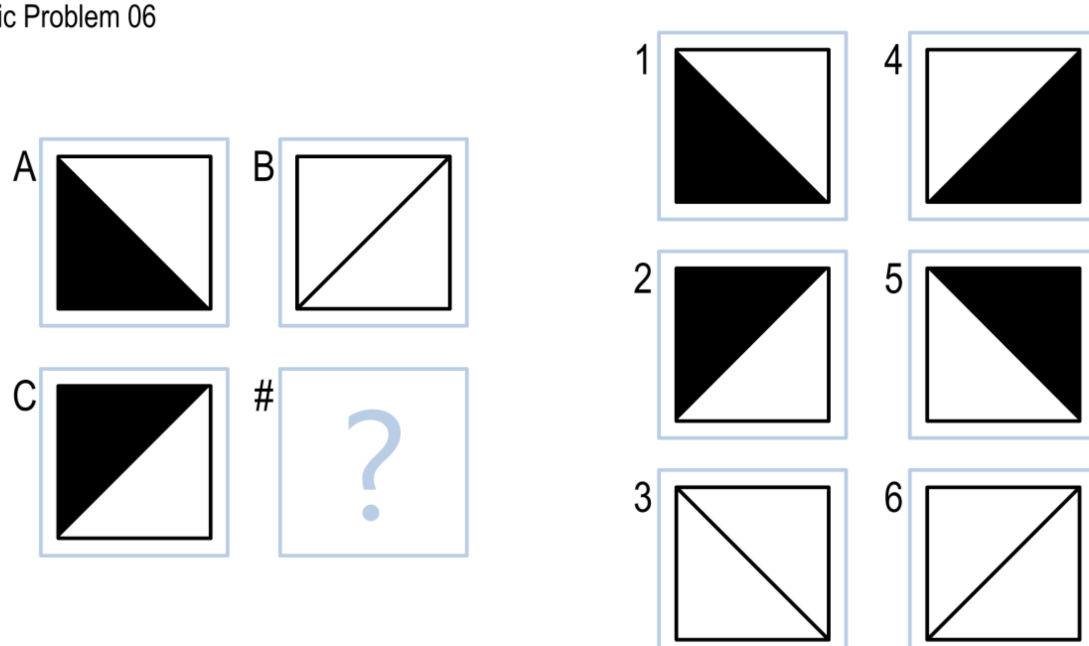
When trying to solve Raven's progressive matrices, the difficult part is programming an agent that can heuristically detect similar problems from the past. Obviously each problem will have structural similarity as figures A, B, and C are always given and the goal is to choose from figures 1- 6. So structural similarity could be useful for an agent with multiple purposes, for it to determine what type of problem it's trying to solve. This could also be applied at a high level for individual Raven's problems, like does it contain multiple objects in each frame or just one? Does every frame contain the same number of objects? Are there any figures that are empty? From these higher-level comparisons, we can limit the state space of which type of problem to map to in memory. Each will also have the exact pragmatic similarity, for in every Raven's problem the goal is to choose a figure from options 1 – 6. So when approaching a situation, the first step for our agent would be to fully analyze the current problem, and apply computational heuristics that will categorize it. These mainly will be done through semantic similarity, by recording not just easy facts about each frame, but what changes are actually occurring in the problem between figure A-C, A-B, etc. Because we have access to the solution for each problem, we can then store the type of problem, but also the fundamental difference and relationship between figures A, B, C and the answer. After this, each needs to be stored in a memory bank that can be retrieved later for help solving a future problem. For example, if facing this problem:

2x2 Basic Problem 02



Our classification heuristic, we could first see that every frame has the same amount of objects and that it's clearly a fill problem. Semantically, we would derive that the answer is 3, and from there we could map the differences as shape and fill. Not just the overall differences, but also the individual differences between each figure. After storing this information, a problem that would similarly map to it would be something like this:

2x2 Basic Problem 06



Because the only difference in this problem is fill, we could use the relationship from problem 3 to determine that the fill relationship needs to be the same for A-C, A-B, and then from C-Answer and B-Answer. It would also determine that the shape changes horizontally from figure to figure as well, just like in problem 2. Using analogically reasoning gives your agent insight and allows them to learn overtime how to more effectively solve problems that they have seen before. It relies mostly on semantic similarity, as that is the most powerful and the best representation for a unique situation.