**Project 2 Reflection**

My agent uses verbal representations to generate horizontal, vertical and diagonal transformations by first ordering the objects from innermost to outermost, left to right most, bottom to topmost, and based on overlap. This can be considered a classification of the different relationships of objects within a figure. Once the list for each relationship class is generated the comparisons between each list of objects is made from first to last. If the sizes of the lists being compared aren’t equal then it is inferred that the object has either been deleted or added. It must be noted, that the transformations are generated based on the knowledge of the variations in the objects bootstrapped into the agent. For instance, using the given attribute that an object goes from “very large” to “large” it will “shrink”.

Once the transformations are generated for the given figures, the goal transformations are generated for the options. If the initial transformations are equal to the goal transformations then the first right solution is selected. This is however not ideal, because we may have a problem in which multiple options may satisfy a solution and it is preferable to identify a best fit in that case. In the next project, the best fit will be found by adapting the algorithm to factor in similarity, i.e. by generating a new solution from the given figures the characteristics of this new solution are compared only against the similar options. The superficial comparison will consist of the number of objects and the object types. If this still doesn’t identify the best fit, a more deep comparison will be made.

Once the best fit is identified the challenge is to identify what knowledge to store so that improved reasoning and learning becomes possible. In this project, no knowledge yet persists from one problem to the next, however in the next project one information that may be useful is expressing in which order the objects are to be compared. The transformations will differ if we change the order of comparison. Consider problem B-10 and its solution 3. If we go from inner to outer we will get “shrunk” and “filled” as the transformation of the innermost object. This is not the same if we go from outermost to innermost for which the transformation is “added” inner filled square, which turns out to be the correct one. Based on varying the order of comparison we can see how many answers we get right so this a case that can be stored with the order and the number of answers that have been correctly answered.

Some of the challenges I have faced with the B problem set was identifying reflection (in B-05) where only an angle is given as an attribute, and the aforementioned comparison order, and future challenges will be identifying new shapes, new transformations, e.g. how can the agent learn that it is a fill without having been told so ahead of time. This should significantly improve the generality of the agent. Unfortunately, due to running out of time, I wasn’t able to do a deep dive into the problems faced with problem set C. The result was abysmal though with 1 correct and the rest either failed due to error or was skipped. Whereas for B, the agent got 7 right, and the rest were skipped due to the problems mentioned earlier. I surmise that the problems with the agent are not fundamental, and should be addressable through fixing/augmenting the existing code and some partial proposition on how to do that was given. I believe that with 3x3 matrices, the agent will have more information to work with hence it should be able to make more accurate inferences.

I chose this method for solving RPMs because it closely resembles how I go about solving the problems. I tend to inevitably make the comparison based on the relationship classes or groups. I thought it would be worthwhile to incorporate that sort of intelligence in my agent and see how it performs.