**Project 3 Reflection**

1. How does your agent reason over the problems it receives? What is its overall problem-solving process? Did you take any risks in the design of your agent, and did those risks pay off?
2. What improvements could you make to your agent given unlimited time and resources? How would you implement those improvements? Would those improvements improve your agent’s accuracy, efficiency, generality, or something else?
3. Which reasoning method did you choose? Are you relying on verbal representations or visual? If you’re using visual input, is your agent processing it into verbal representations for subsequent reasoning, or is it reasoning over the images themselves?

For the 3x3 case, my agent uses problem reduction to start with a 2x2 sub-problem by considering only frames E, F and H to generate a solution with the assumption that these frames contribute the most useful information to the problem. This is a risk I took, because it is after all based on a conjecture after studying all the problems. I knew also that a problem such as C-07 is easier solved by incorporating more frames. It didn’t really pay off however, because the number of questions my agent answered correctly improved by only 1 for Basic Problems C.

Frame transformations are generated for EF, EH, FI and HI where I is one of the options given. The transformations are generated by first ordering the objects based on relative position, i.e. top to bottom, inside to outside, left to right, and overlap all of which are stored in a hash table. By having consistent ordering we are now able to compare the attributes of what are thought to be corresponding objects. Based on the comparisons, classification of the transformation is generated at the object level and then more abstractly at the figure level. These classifications consist of no change, move, multiply, add, delete, resize, shape change. After the transformations are classified, I then iterate through the transformation frames to identify if the classifications match. The solutions that have a match are cached and if there is no match the problem is skipped. (In the future, for any problem that results in a “skip” I will modify the current code to process the next 2x2 sub-problem which is A, C, G, and I. ) Then the next step is to iterate through each of the object transformations and see if they are all equivalent. If not then, there is one more comparison that is made which is to identify if there is similarity in each of the objects. If these reasoning methods allow us to eliminate the possible solutions and we are left with one solution, then that solution is chosen otherwise we skip the problem. (As mentioned earlier, in the future instead of skipping we will create another 2x2 sub-problem and try to solve that until we arrive at a solution. This should in general improve the accuracy since we are able to use more information to narrow in on the solution, however if we still have multiple solutions then we will select one and check the answer. If the answer is incorrect we will use the true answer to compare the classification of our answer and the true answer. If the classification is different than we will retrieve the classification of another 2x2 sub-problem and compare its classification, until we arrive at the same classification. We will then assign higher confidence to this 2x2 sub-problem and start with this sub-problem for the next problem. This should make our agent more efficient since over time we should arrive at a solution faster.)

My agent currently only uses verbal reasoning. I wanted to improve the accuracy of the verbal reasoning and so continued to work on that in this project as well. I have noticed that the multi-pronged method I introduced in this project helped improve the accuracy of both Basic Problems B which improved from 7 to 9 and C which improved from 1 to 2. I will continue making the improvements mentioned earlier going forward and I expect to see generality, accuracy and efficiency improve as a result.

The way I intend to incorporate visual reasoning is by performing the translation of visual to verbal reasoning. Through visual reasoning my agent needs to be able to detect the objects and their attributes and then generate the text files that will contain the object information. From there on the agent will essentially function as before. The object recognition will be achieved through constraint propagation as discussed in our class lecture.

1. How does your agent actually select an answer to a given problem? What metrics, if any, does it use to evaluate potential answers? Does it select only the exact correct answer, or does it rate answers on a more continuous scale?

There is no metric used per se, however a multi-pronged approach is used to narrow down the possible solutions as mentioned earlier. Through the multiple steps that are executed answers are eliminated until we arrive at one solution.

1. What mistakes does your agent make? Why does it make these mistakes? Could these mistakes be resolved within your agent’s current approach, or are they fundamental problems with the way your agent approaches these problems?

There are several mistakes my agent currently makes. The ordering of objects based on relative position only works for one set of objects, i.e. if they are top to bottom, left to right, etc., however the algorithm doesn’t yet take into account having more than one set of concentric circles or other types of object relationships. It also doesn’t take into account a figure that has nothing in it, in which case there are no objects to generate a transformation. This will likely cause problems. Also, the agent already has bootstrapped transformations, however is not able to learn new transformations. The first two mistakes can be fixed however the last one involves learning which is much more difficult to solve. It will however help improve the agent’s generality.

1. How well does your agent perform across multiple metrics? Accuracy is important, but what about efficiency? What about generality? Are there other metrics or scenarios under which you think your agent’s performance would improve or suffer?

I would say that my agent is somewhat efficient at solving the RPM. The current running time is less than a second, however if we implement what is planned for the future it is quite possible that the performance will degrade. This is mainly because image processing, and processing other 2x2 sub-problems will have additional overhead. In addition to that if we have a much larger set of objects the memory usage will be high mainly to take into account the additional transformations.

My agent is primarily bootstrapped with the type of object transformation that can occur and has a predefined set of classifications. If the agent encounters a new transformation it will not be able to classify it. This means that my agent is not very general. I don’t have a metric for generality, however this can be easily tested through a scenario like the test problems that will be used by the TAs. By giving problems that are found in the world, generality can be measured. The results of that should directly influence the evolution of the agent.

1. Finally, what does the design and performance of your agent tell us about human cognition? Does your agent solve these problems like a human does? How is it similar, and how is it different? Has your agent’s performance given you any insights into the way people solve these problems?

My agent has improved in performance for the most part over the course of the projects and if I implement the changes that I have discussed and correct its mistakes, I feel very confident it will do well. Having said that, my agent currently doesn’t solve the problems like a human does. Take for instance C-07. It took me a second to glance at the big picture and identify the solution. I didn’t follow any predefined steps to arrive at the solution, it was primarily all visual. I could detect the pattern visually with minimal thinking. This is human cognition in the visual form whereas in C-03 I was using my understanding of addition to arrive at the solution. So it appears human cognition has many forms. My agent doesn’t follow this approach, rather it starts taking a subset of the problem given, analyzes it and then tries to arrive at the solution. If it’s not able to then it tries to eliminate the possible answers by looking at more subsets until it arrives at a solution. In order to solve the correspondence problem it tries to order the objects in terms of relative position and then compare against another figure objects that are ordered. This approach is similar in human cognition however I would argue that it is not as granular. For instance in C-04 I found myself counting the circles, however I didn’t order them. My understanding thus far is minimal, however I believe there are many forms of human cognition and it is sometimes difficult to replicate.