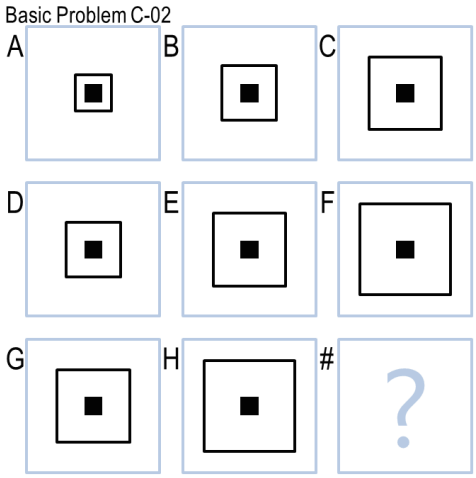
xxiao39@gatech.edu

Project 2- Project Reflection

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?

In Project 2, we need to further solve 3x3 matrix problems in which the rightmost cell of the bottom row is missing by picking up an answer from 8 options.
Solving these problems essentially figures out the underlying rules that explain transformations among cells. Here is an example:



There is a very simple rule that similar transformations exist between adjacent columns. For example: A->B most likes B->C, D->E most likes E->F, G->H most likes H->?
Based on this rule, my agent reasons over images first. And my agent also uses generate & test method with semantic network which is a common tool used to solve ravens problems.
My agent obtains a transformation value between two images by getting the difference of two images' pixels. In java, getRGB(weight, height)function can get the value of color pixel. My agent has gotten the transformation value between G->H. The value infers the mapping between H and ?. Then my agent obtains a candidate answer.

difference of Pixels(G->H) $----closest---\rightarrow$ **difference of Pixels(H->?)**

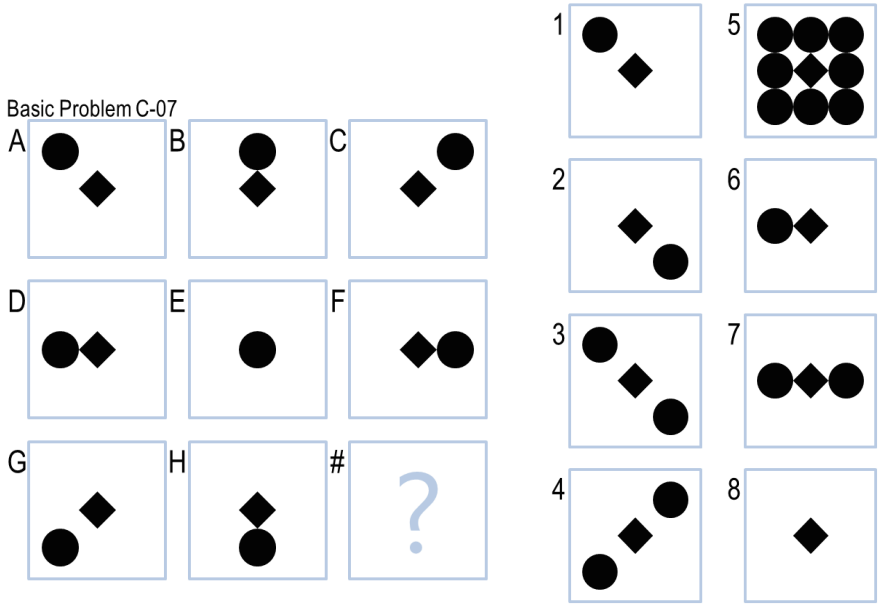
Only image comparison is not perfect. Next, my agent will apply verbal representation. And my agent applies another rule that constant transformations between first cell and third cell in every row. As we know, there are basic transformations between two images. My agent lists some basic transformations that are identity, mirror, flip, rotate90, rotate180 and rotate270. These transformations meet most common situations. My agent uses Generate & Test with Semantic Network method. My agent obtains transformations between A and C in the first row by comparing attributes between them from verbal information. The transformations infer the mapping between G and ?.

transformations(A->C) – – – – – –→ **transformations (G->?)**

In the design of my agent, my agent cannot figure out a correct answer especially when only orientation of objects is different. But combined with verbal comparing, most of the errors have been corrected.

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

Let us see the problem below.



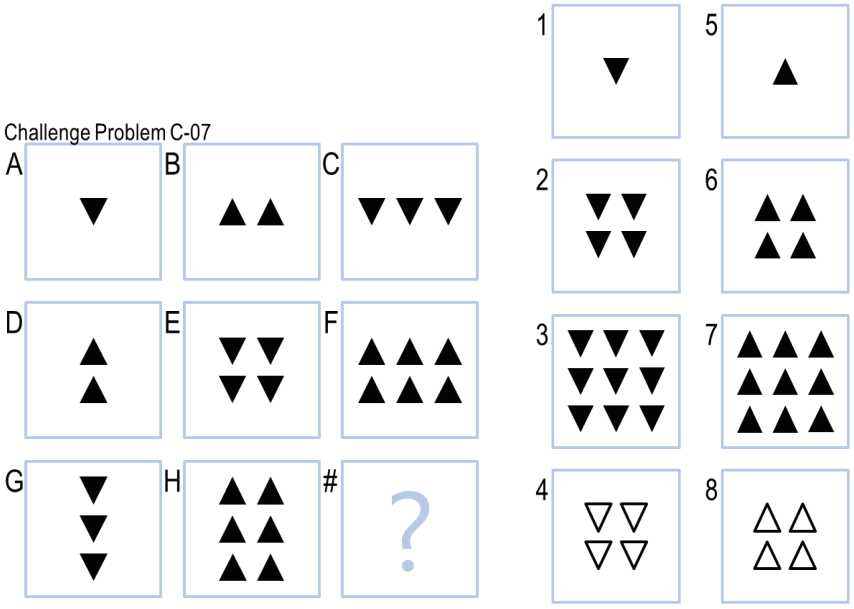
My agent first uses visual method by comparing the difference of image pixel between G and H to the difference of image pixel between H and ?. My agent gets an answer(option6) which is not correct. We find option2, option1 and option6 could have similar pixels. My agent is hard to identify which is correct by only visual method. But it is easy for my agent to provide a list of candidate answer indices based on visual method. For example answer[1,2,6]. Then my agent can make a judgment whether A->C is identity relationship or mirror relationship or flip relationship or rotation relationship by verbal method. If my agent know the mirror relationship between A and C, it can infer the very similar relationship between G and ? and provide a correct answer. My agent gives priority to the answer from verbal method because verbal method uses more specific information.

mirror(A->C) – – – – – –→ **mirror (G->2)**

My agent selects only the exact correct answer. It does not rate answers on a more continuous scale because the agent does not score the transformation.

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

My agent can quickly and correctly solve 12 basic problems. These problems generally have very common transformations. For these problems, verbal and visual methods are good enough. But when my agent tries to solve 12 challenge problems, only 6 answers are correct. These challenge problems have no verbal representation and only visual representation, so my agent only implements visual method. Now this visual method is too simple to complex problems. My agent cannot get the detailed information about objects in images. Let us see the following problem.



My agent gets an answer(option7) which is not correct. The correct answer is option3 where the direction of triangle is downward. However in option7 the direction is opposite. My agent cannot catch the direction information of objects in images. My agent's current approach cannot resolve these mistakes unless my agent has verbal information. In reality, my agent will meet more complex problems. When my agent solves these problems, my agent must strengthen images processing capabilities.

What improvements could you make to your agent given unlimited time and resource? How would you implement those improvements? Would those improvements improvement your agent's accuracy, efficiency, generality or something else?

Given unlimited time and resources, I make much improvement to my agent. My agent has limited verbal and visual methods. First I need improve verbal methods. My agent is designed to handle problems that have basic transformations. I plan to add more common variations as production rules. For often- encountered problems, the agent has more production rules and can provide the right answer in a short time.

1.Orientations of one or more objects are same in each row or each column, but changes down a column or right a row.

2.An increase or decrease between adjacent cells in number or size. An image from one row or one column is added,

subtracted or overlapped to another image from same row or same column to produce third image.
3.Three values of a category such as shape and color are always present in one row or one column.
Second I need improve visual method. I want to add affine method to visual method. The affine method assumes elements with a row or a column are related by similitude transformations. It uses a model of similarity.

Similarity(A,B)=$f(A \cap B)/(f(A \cap B) + \alpha f(A - B) + \beta f(B - A))$

It discovers which similitude transformation best fits any of the complete rows or columns, applied this transformation to last row or column to generate a candidate answer.
Third I want to build a database to store past cases.
My agent first can apply generate & test on problems, then check to see if the answer is right. If not, it might store that problem as a case in the database. The on future problems, my agent might find a close match between new problem and old case in the database. My agent can apply that case's reasoning instead of generate & test method.
Definitely, I think if these improvements are implemented, my agent's accuracy, efficiency and generality will be improved greatly.

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?

My agent did very well across multiple metrics. For 12 basic problems, all answers are correct. So my agent has very high accuracy. My agent uses some very common production rules, such as similar transformations between adjacent columns; constant transformations between adjacent cell in each row. My agent only considers very limited production rules. So my agent has limited generality.
Because of lack of a long-term memory, my agent's efficiency is not outstanding. In order to improve my agent's performance, other production rules also should be considered. As I mentioned above, orientations of one or more objects are same in each row or each column, but changes down a column or right a row. An increase or decrease between adjacent cells in number or size. An image from one row or one column is added, subtracted or

overlapped to another image from same row or same column to produce third image. Three values of a category such as shape and color are always present in one row or one column. And my agent mainly focuses on the row and column correlation, my agent need pay attention to diagonal correlation. In some complex problems with more than one rule, my agent must figure out which elements are governed by the same rule.

Which reasoning method did you choose? Are you relying on verbal representation or visual? If you're using visual input, is your agent processing it into verbal representations for subsequent reasoning, or is it reasoning over images themselves?

My agent applies Generate & Test with semantic network method to ravens problems. Semantic network provide a level of abstraction at which the problem gets represented and analyzed. Knowledge representation ignores things that are at a low level of detail to avoid the problem too complex. Calculate a transformation value from G to H. My agent finds out which one of transformation values (H->?)is closest with the transformation value from A to C. Further analyze and get the transformations from A to C. My agent finds out which one of basic transformations (G->?)is most similar to the basic transformations from A to C. Then the agent will get the right answer. My agent relies on both verbal representations and visual representations. My agent can extract attributes of objects and compare these attributes from verbal representation. By comparing, my agent can infer some basic transformations.
My agent cannot process visual input into verbal representations, but it can reason over images themselves. White pixel and black pixel are different. So they return different values. Every object is made up of pixels. My agent accumulates pixels in images and obtains a total image pixel value. The difference of two image pixel values reflects some transformation. But it cannot figure out specific transformations. The single dependence on visual representations is not good enough. My agent tries to choose to combine verbal and visual representations.

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

Everyday we often use Generate & Test with semantic network as the problem-solving method. We cannot have a complete and correct knowledge system. We only have limited computational resources. For different scenarios, we do not always have resource to ensure the inference is correct. So we need come up with potential solutions to a problem, and test the solutions out.
My agent uses observations from verbal and visual representations and forms basic transformations. Then the agent compares these transformations and tests them. The answer that has the closest correlation is picked up. A human does a same procedure when he solves a problem. But a human with a long memory, good intuition and good creative ability has deeper knowledge and better analysis ability. My agent has not.
My agent gives me many insights into the way people solve these problems. It makes me understand deeply how we think and reason using knowledge when we try to solve complex problems.