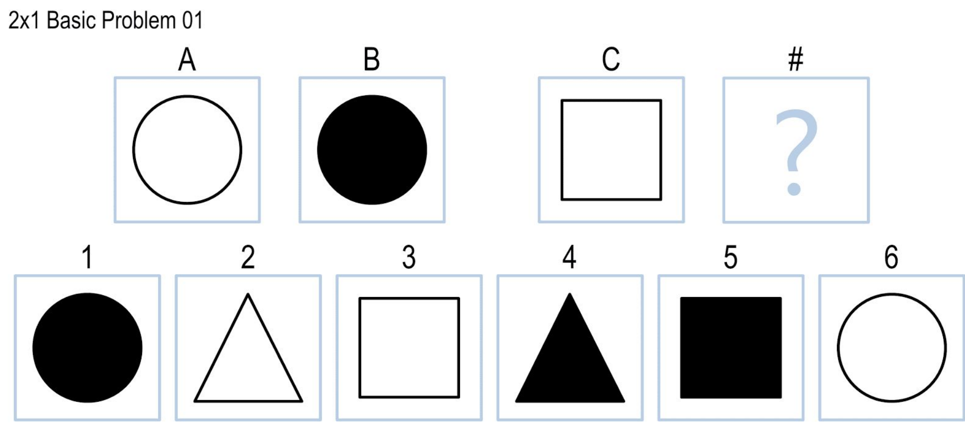
Corey Crooks
ccrooks6@gatech.edu
01/18/2018
GT ID: 903130397

Addressing Raven's Progressive Matrices

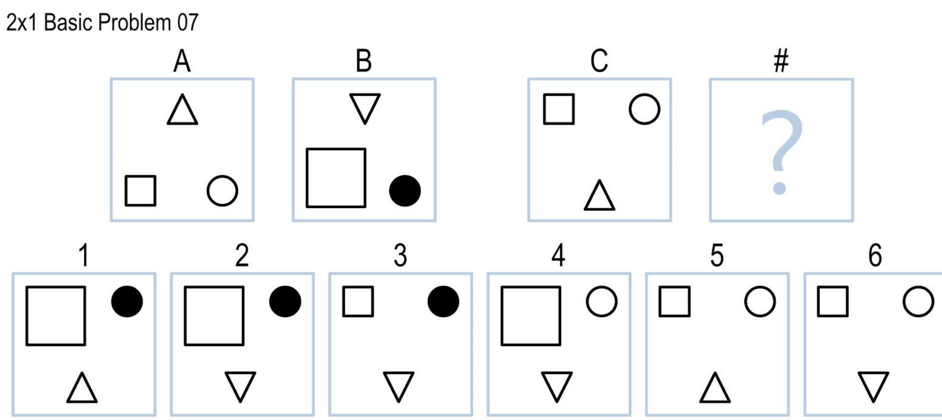
Ravens Progressive Matrices (RPM) is an intelligence test typically used in educational settings. It is usually comprised of visual multiple-choice questions which gauge the testers ability to reason and problem solving ability. In order to measure intelligence, RPM focuses on two distinct components: the ability to think clearly about complex ideas and the ability to store and recall information. RPM was originally developed by John C. Raven in 1936 where the subject is asked to identify the missing element that completes a pattern. These tests are presented as patterns in the form of 6x6, 4x4, 3x3 or 2x2 matrix, giving the test its name.

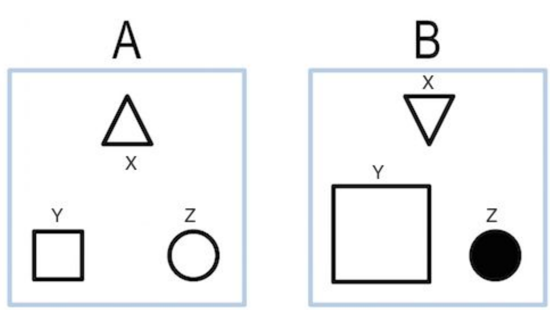


When actually addressing a Ravens Progressive Matrices problem it is probably best to start with a basic example first. The following image (2x1 Basic Problem 1) shows two groups of lexical nodes A,B,C,#(or D) along with 1,2,3,4,5,6. The goal of this problem is to pick the correct number that solves the transformation from C. The idea is to analyze the transformation of A to B and apply that to C. Once that transformation is found then the case can be made for the correct answer 5. This problem is very simple to solve, however the real question is how can a computer solve such a basic problem?

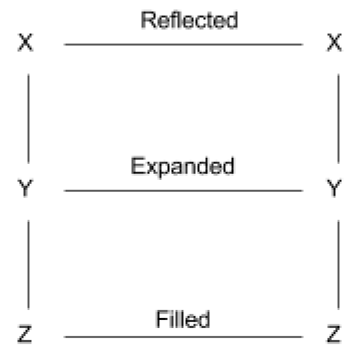
While continuing to use the example from above, the answer can be almost seen instantly. There is a round circle that becomes filled such that for C, the square must become filled in as well. When trying to convert this type of language for a computer to understand could become a very tedious task even for basic problems. Visually us humans can make the comparison, however in the computing space we find that there are many more variables in the sector such has x, y and z coordinates for location and depth, length, space and other various factors that all need to be accounted for to solve such a problem. These variables will become even more difficult as the difficulty in problems increase.

There are two main problem solving techniques that will help computers solve ravens matrices tests which are Semantic Networks and Generate and Test. The concept behind semantic networks is the structure which consists of lexical nodes (x,y,z,), directional links (arrows) and application-specific labels. In the basic problem 7 example, the first concept is to apply lexical nodes to each box starting with A.

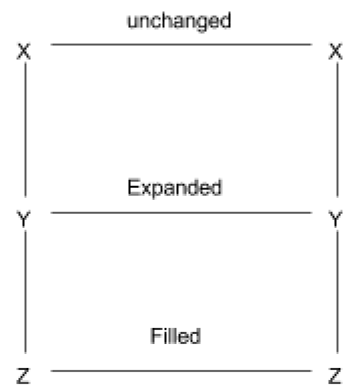




Once the three lexical nodes have been applied, the relationships can then become identified using directional links and application specific labels as seen below.



Once the transformation is found between A to B using semantic networks, the same can apply to C with every option available for the answers section. For example, If applied to C and 1 the following would result in:



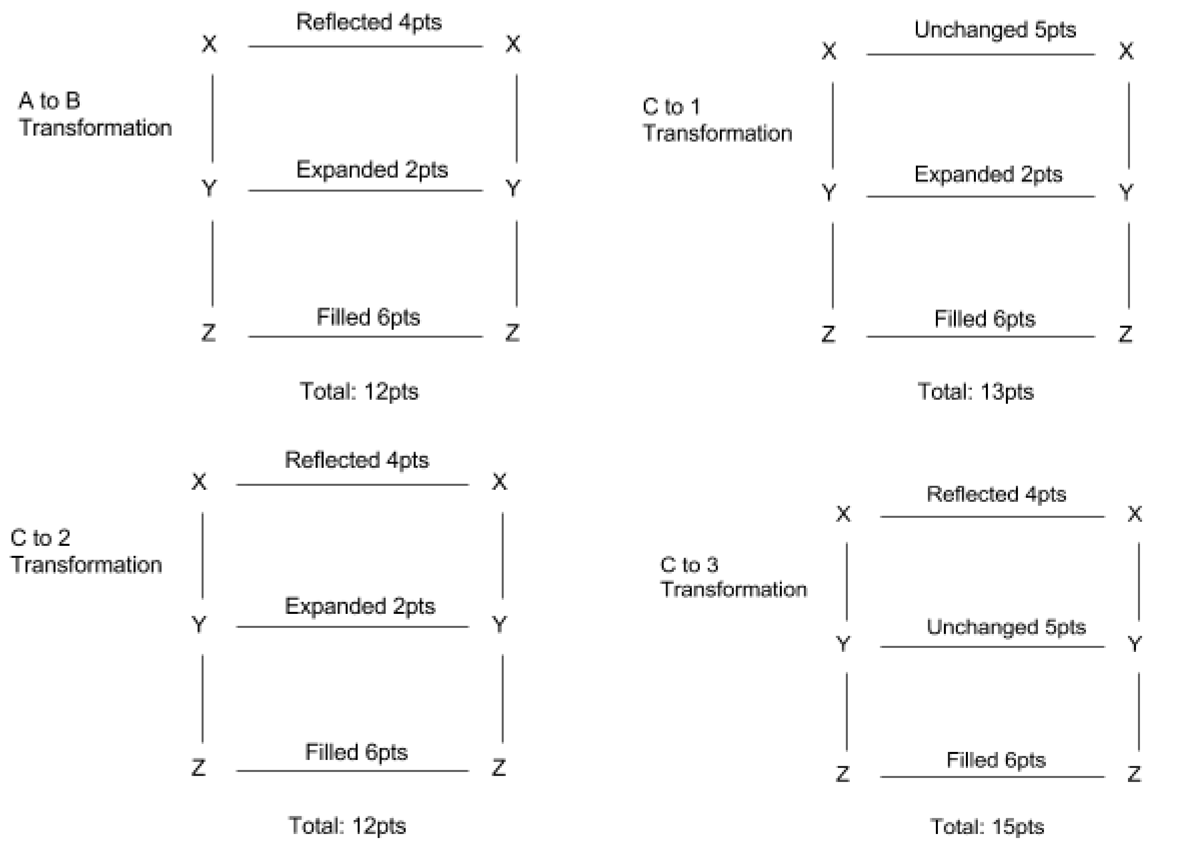
After applying this transformation from C to 1 this is verified to be incorrect. However if you re-apply the transformation to answer 2, the answer will match the transformation between A to B. This method is a very good method to use when solving such problems however there is a way to improve upon it by introducing similarity weights.

Similarity weights can add a more distinct way of finding out the answer to a problem by attaching weights to each transformation. Here is a simple table that correlates the weights to specific transformations:

6 points	Filled
5 points	Unchanged
4 points	Reflected
3 points	Rotated
2 points	Scaled

1 points	Deleted
0 points	Shape Changed

While still using the basic problem 7 the newly added weights for the transformation of A to B are as followed:



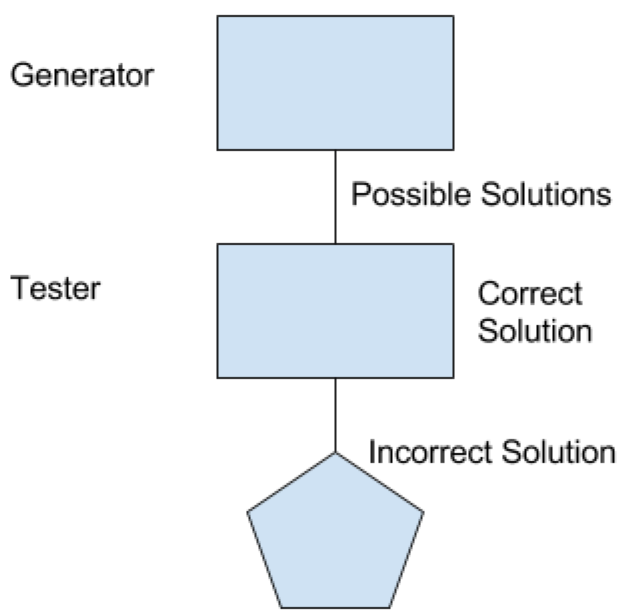
Transformation				Totals
C to 1	2	6	5	13
C to 2	2	6	4	12
C to 3	5	6	4	15
C to 4	2	5	4	11
C to 5	5	5	5	15

C to 6	5	5	4	14

Transformation				Totals
A to B	2	6	4	12

After completing all of the transformations from C to 1 - C to 6, the answer can be pinpointed to number 2. When weights are added to the transformations, better accuracy in predicting the correct answer follows. Weights will help the agent continue to solve problems along with more advanced, difficult ones as well.

The next problem solving technique is Generate and Test. Generate and Test comes from the fundamental idea of having a problem then generating potential solutions and testing them. Potential solutions that need to be generated vary depending on the kinds of problems. For some of the problems, the possible solutions may be particular points in the problem space and for other problems, paths from the start state. Generate and Test is a powerful problem solving method when paired with semantic networks.



1. Generate a possible solution
2. Test to see if this is the expected solution

3. If the solution has been found quit else go to step 1

There are two different ways of applying Generate and Test to Raven's matrices. The first idea is to distinguish the transformation between A and B like previously used in semantic networks. Once the transformation is completed, apply that transformation from C to D. Once the transformation is applied for D, compare it (test) to 1,2,3,4,5,6 to find a close match. If a transformation is not found during the testing or it doesn't meet a certain level of confidence then go back and remake the transformation and repeat the process.

Another way of applying Generate and Test is the following:

1. Start at first answer in problem set and put into slot D
2. Generate transformation from C to D using semantic networks
3. Generate Transformation between A and B
4. Test transformation between C and D to A and B
5. Does C to D match close to A and B?

    Yes? Exit.

    No? Increment answer number, go to step 1.

As previously stated, Generate and Test along with Semantic Networks are very powerful when used in unison. In order to develop an agent smart enough to solve basic ravens matrices tests along with even more challenging tests, one must utilize both of these problem solving techniques. The idea of building such agent comes with many challenging development tasks. When given the input text file, all visual representation is stripped away and the agent must solve each test through objects and their descriptions. These descriptions are the most important aspect when building the semantic network along with the basis of Generate and Test.

In conclusion Raven's matrices problems are tests to challenge our reasoning and problem solving ability. There are two different ways of being able to solve these tests through Semantic Networks and Generate and Test. These two problem solving techniques are critical to our agent when it comes to figuring out the correct answer on its own. The agent will have to use its pre addressed knowledge that it had previously seen and apply it to figure out the answer with a certain level of confidence. Ravens Matrices are a very interesting set of tests that will not only challenge humans ability to reason but also an agents ability to think like a human.

Sources:

https://sites.psu.edu/psych256sp15/2015/04/19/problem-solving-generate-and-test/

https://classroom.udacity.com/

http://www.coursedocs.org/gatech/cs7637/03---semantic-networks.html

https://www.alleydog.com/glossary/definition.php?term=Raven%27s%20Progressive%20Matrices%20(RPM)