Solving 3x3 RPM (Visual Approach)

Varun Dani (vdani3)

November 2017

1. Abstract

This report address solution of Raven's Progressive Matrices (RPM 3x3) tests, based upon only visual representation. First It describes difficulties with both the verbal approach and combination of visual and verbal approach for Project 1 and Project 2. After that, overall architecture, design, and enhancements are described with challenges encountered and performance evaluation of agent. Finally, describes agent's cognitive connection to KBAI and human intelligence.

2. Project 1 and Project 2 Details

2.1. Project 1 Overview: Design and Overall Reasoning Strategy of Agent

Implementation of Project 1 had followed the 1st principle of KBAI "Agents organize knowledge into knowledge structures and use this knowledge to guide reasoning." (Ashok & David,2016, L2 video:17). The Agent imitates frame like structure from verbal representation (Detailed in Architecture) for knowledge representation for different property/value of Object in Figures.

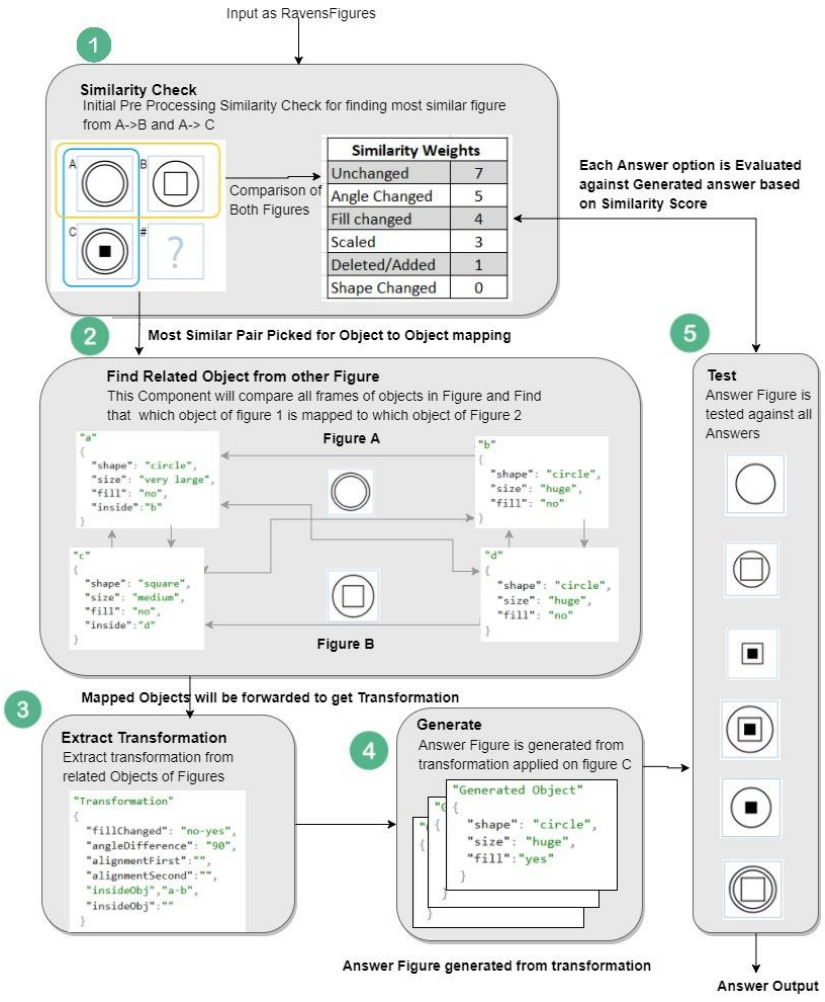


Figure 1: Overall Architecture for Agent (Goel Ashok 2017, Basic Problems p.10 f.1) (design tool draw.io)

2.2. Project 2 Overview: Design and Overall Reasoning Strategy of Agent

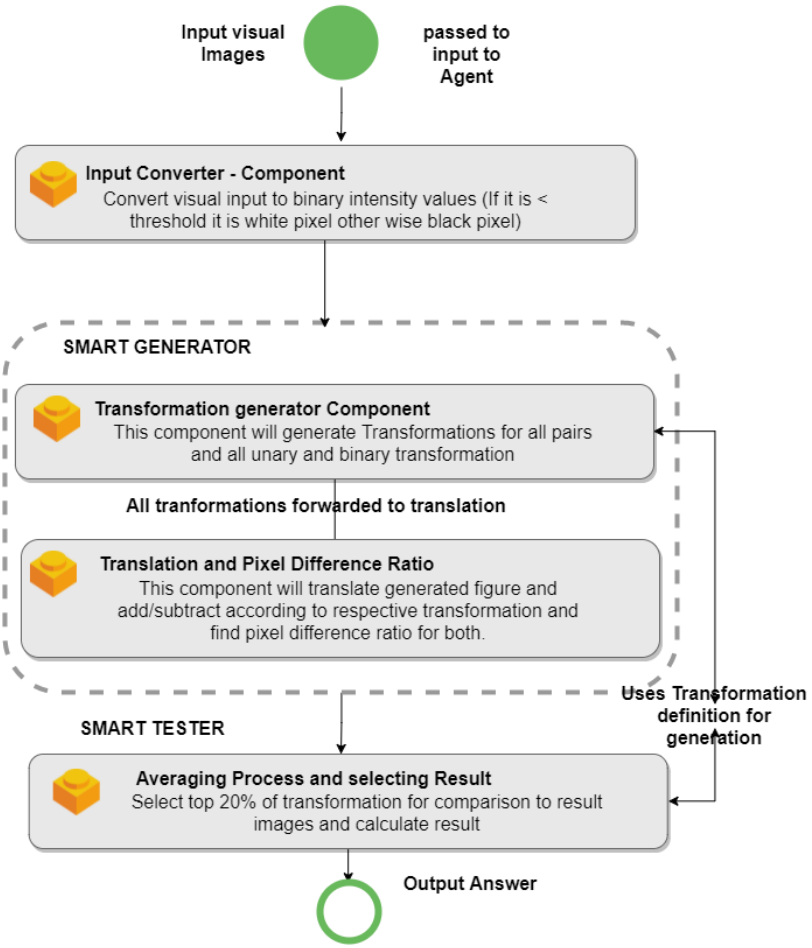


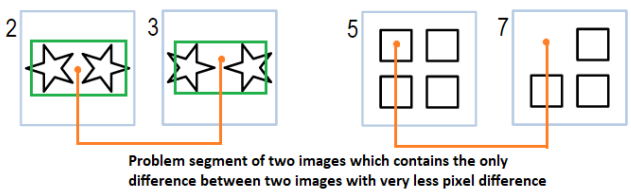
Figure 2: Flow Architecture and Design of Project 2(design tool draw.io)

2.3. Challenges and limitation of Agent in Project 1 and Project 2

Scaling of Ontology for transformation: With various figures and scales, in pure verbal approach, it would be very difficult to maintain and understand ontology and their mapping for each object.

New Added or Deleted Objects in Figure: For only verbal approach, it will be very difficult to track one object in one figure to another object in another figure. For Example, if a square changes shape to a pentagon, the agent will consider as deletion of square and addition of pentagon as transformation.

Selecting relevant Answer from mostly similar images: Because of ambiguity in answer images, the agent tends to select wrong answer image based on only transformation and pixel counts.



In conclusion, I have observed issue of scalability in agent's design and verbal approach and accuracy issue in the agent's design for selecting ambiguous choices. Following is Reasoning strategy for agent's design in Project3.

3. Reasoning Strategy of Agent and Improvement on problem sets

Reasoning strategy of the agent is inspired by Hunt's (Hunt 1974) Gestalt Algorithm and the Computational model for solving RPM (Kunda, McGreggor, Goel, 2013). Reasoning strategy includes following basic steps:

3.1. Finds the most similar figures from given set of images in the matrix. This will be carried out by finding maximum similarity from each comparison described as following. The most similar comparison and transformation are marked and forwarded to next step.

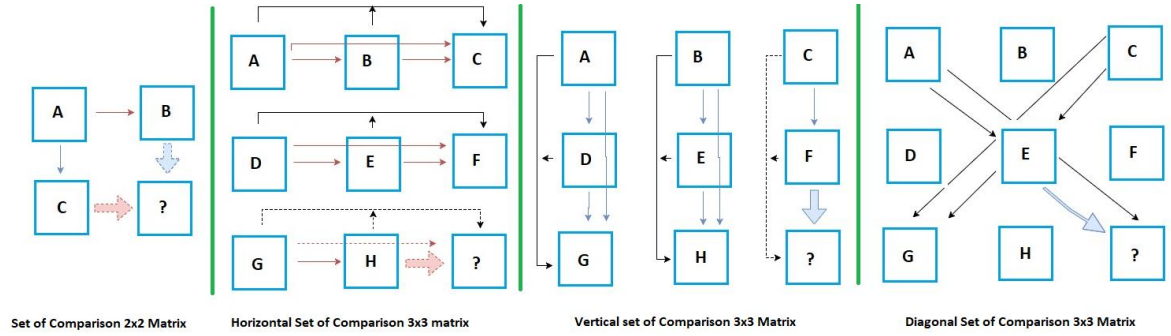
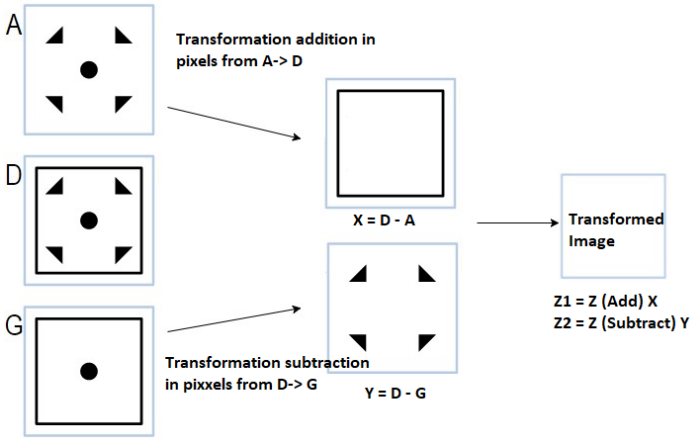


Figure 2: Transformation Pairs (Kunda, McGreggor, Goel 2013) (design tool draw.io)

3.2 Extract transformation from given most similar figure. This will be used to transform the respective image in subsequent steps.

3.3 Apply this transformation to respective matrix image to get transformed Images and also apply pixel differences to know exact difference between two figures.

3.4 Learning from transformation: The agent calculates difference matrix by respective addition or subtraction of images and apply this learning to tweak final generated transformation.



3.5 To get accurate answer from ambiguous answers, the agent gives reward based on similarity and outputs answer on basis of highest reward and similarity.

Backward compatibility: Mentioned reasoning strategy will work on 2x2 matrices because of transformations are visually dynamic and the agent considers only unary transformation while executing 2x2 matrices. Also, for 3x3 matrices in problem set D, there is 10% degradation in the basic set from previous reasoning strategy because, in this strategy, the agent will be more generalized based on all problem sets.

4. Design and Architecture of Agent with visual capabilities

Description and responsibility of all component for determine answer image

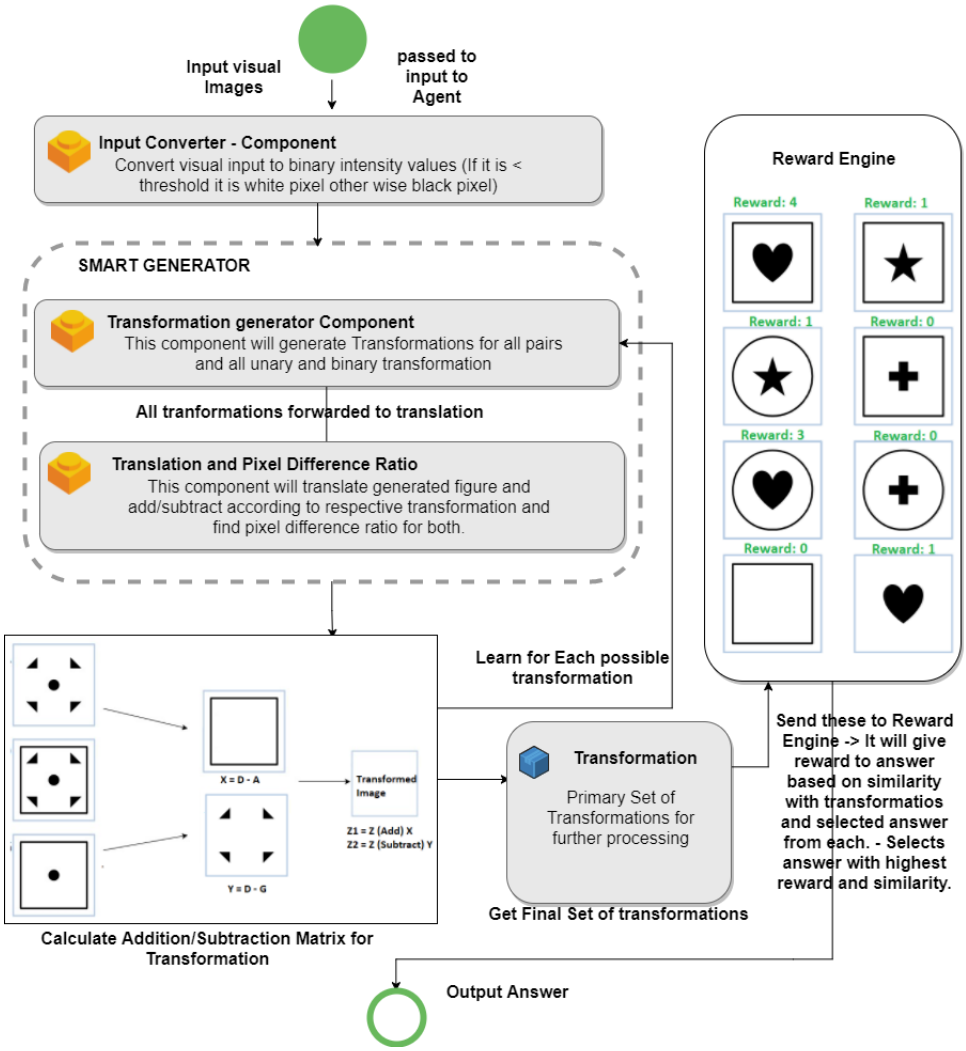


Figure 1: Flow Architecture and Design(design tool draw.io)

1. Input Image is passed and converted to binary image for further transformations. Pixels < threshold values are converted to white pixels(0) otherwise black pixels(1). The reason for taking black pixel as 1 is that for subtraction of image it will keep that pixel rather discarding in subtraction.

2. For all pairs of figures following transformations will be applied and similarity values calculated from comparison of transformed images and respective image.

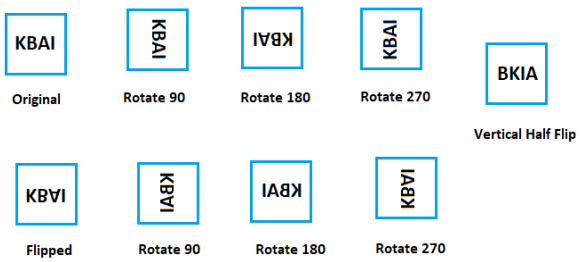


Figure 3: Unary Transformations (Kunda, McGreggor, Goel 2013)

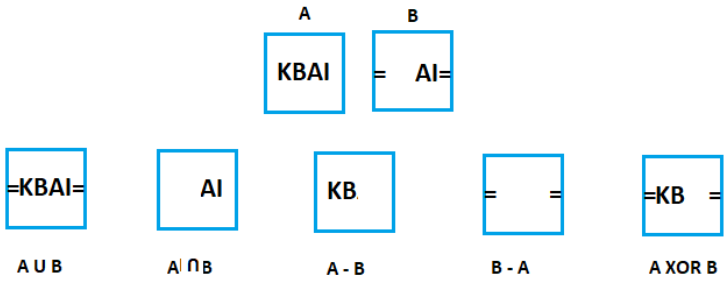


Figure 4: Binary Transformations (Kunda, McGreggor, Goel 2013)

3. Similarity and pixel ratio calculated for all transformation that are generated from combination of all. Calculation of Similarity is captured from measure from Tversky's (1977) ratio model of similarity. Performing this equation will give us similarity of performed transformation and given figure. $\text{similarity}(A,B) = \frac{f(A \cap B)}{f(A \cup B)}$

   Also, if it is 3x3 matrix, pixel ration from figure A and B is calculated as **D = (# of pixels in Image A)/ (# of pixels in Image B)**

4. Smart Tester will pick most similar transformations and apply to get similarity values. Then generated transformation tweaked according to transform matrix X or Y. For example, for 3x3 if AB-> C then applies to (GH) & (transformed Matrix)->Answers. After that, all tweaked transformations compare to answer images and give the reward to answer images based on highest similarity. The output will be answered with highest reward points and similarity.

5. Challenges and Limitation of Agent

5.1 Priority of Applied Transformations

While selecting transformations for testing with answer choices, It got sometimes difficult with similarity to select exact transformation for given figure.

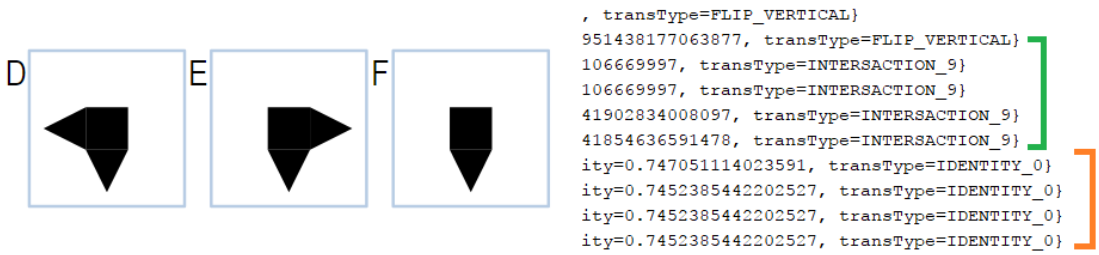


Figure 5: Priority of transformations and Sorted listing (Goel Ashok 2017, Basic Problems p.14 f.1)

For example, for given figure, it will be difficult for the agent to select which transformation to apply for the matrix. One possible intermediate solution is to take maximum threshold, but this approach is limited to the threshold. Also, another improvement is giving rewards by iterating all transformations and this has drastically improved agent's performance but again it is bound by the limitation of only 80% generalization over all problems.

5.2 Accurate answer selection on extremely minor change

Because of very low difference in answer images and ambiguity, the agent tends to compromise on extremely finer details of Image.

For example, below figures contain two answer images that have minor details in pixel differences. This ambiguity leads agent for selection of the wrong answer. For the second image, the agent is capable of identifying difference from design but for first it is not because of very minor change in fragment of section.

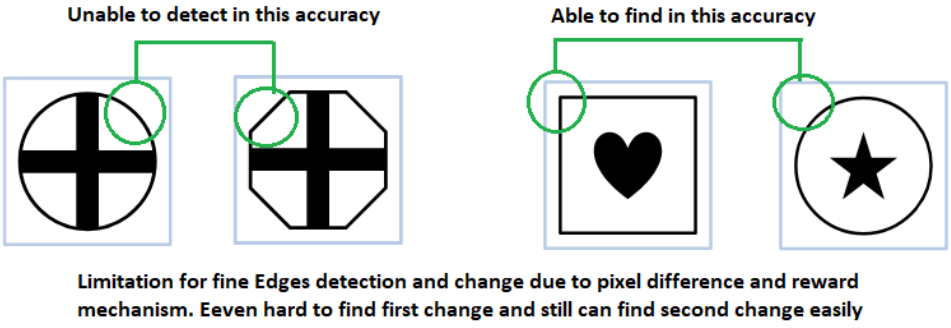


Figure 6: Answer Images with very minor difference creates ambiguity (Goel Ashok 2017, Basic Problems p.20,21 f.1)

6. Performance of Agent

6.1 Performance Analysis on Problem Sets:

Agents Performance on Basic and Test problem set D,E and impact on set B and C.

	Basic Problems	Test Problems	Raven's Problem	Improvement %
Problem Set E	10 <- 12	8 <- 12	9 <- 12	NA
Problem Set D	8 <- 12	5 <- 12	4 <- 12	NA
Problem Set C	8 <- 12	7 <- 12	6 <- 12	**10%**
Problem Set B	12 <- 12	11 <- 12	10 <- 12	**65%**

Figure 8: Performance Comparison of Agent (Problem Set B,C,D,E)

6.2 Computational Complexity and Space Complexity Analysis:

Following are the measure of time and space complexity of Agent in terms of finding the answer of one figure. (For one iteration).

Agent converts all images to the 2D binary array with a singular intensity value for each pixel in array space. Space complexity is **S = O (# of pixels in one Image^2).** We can consider # of figures as constant that varies according to problem set (for 2x2 constant = 9 and for 3x3 constant = 16)

Time Complexity of Agent is dependent on input problem set and transformations applied on them. In general, time complexity will be **T = O (# of transformations * # of pixels^2) + O (# of transformations * 0.2).** For 2x2 problem set, # of transformations are only singular transformation and for 3x3 problems, # of transformations are unary as well as binary).

Also, time complexity of reward mechanism also be in constant time for one problem as it is using constant array space and computations.

For N number of problems, time and space complexities are O (N * S) and O (N * T) respectively.

6.3 Performance Improvements in accuracy based on reward:

This performance improvement in agent's design is after addition of reward mechanism for each answer transformations. To overcome the problem of ambiguous transformations in project 2, each answer choice will get reward point based on transformations. As a result, we can easily identify exact answer from two ambiguous results. For Example, for basic problem D-12 it is first ambiguous between answer (3) and (6) but it is clearer after these improvements.

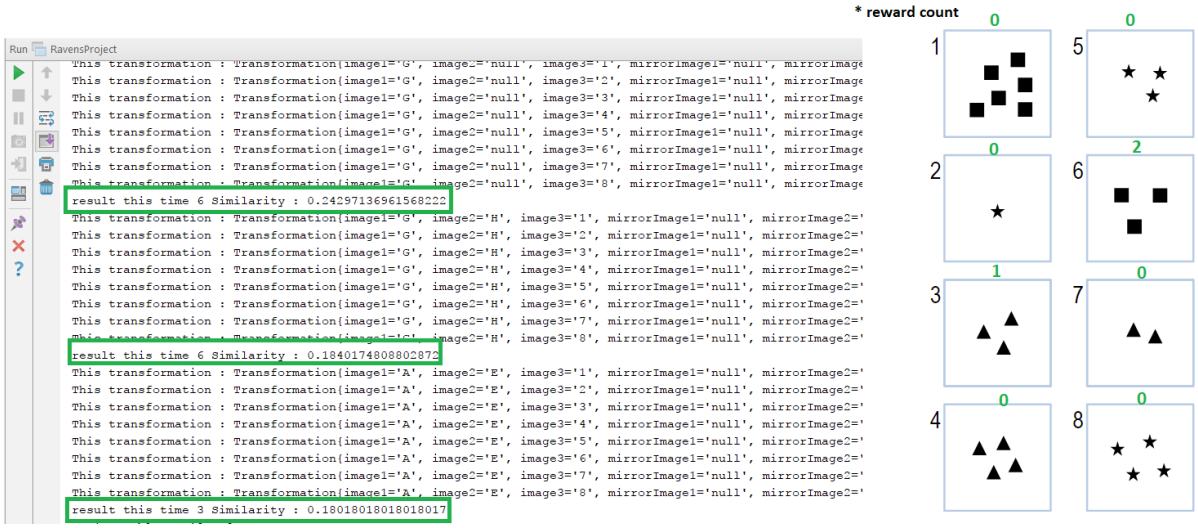


Figure 6: Answer Images with very minor difference creates ambiguity (Goel Ashok 2017, Basic Problems p.36 f.1)

7. Cognitive Connections

The agent uses Generate and Test Methodology of KBAI with the smart generator and smart tester. The smart generator learns and generates transformed Image from best similar transformation and Tester tests similar outputs for threshold limit of transformations and rewarding each transformation to take the decision of answer.

Learning: The agent learns transformation from the best match (horizontal, vertical, diagonal) and applies this knowledge to generate respective result transformation. In summary, we can say that the agent is implementing case-based reasoning inside the problem. It learns from different transformation within problem, stores it as cases, and apply these cases with the little tweaking to get answers.

"*Cognitive control and value-based decision-making tasks appear to depend on different brain regions within the prefrontal cortex*"*(Jan Glascher, 2015)* The human mind will make the decision among choices by Risk and Reward mechanism process in these regions *(Ming Hsu, 2015).* The human mind gathers knowledge from multiple sources, compares and analyse this knowledge to reach the final conclusion of given problem. The agent imitates human behaviour by giving reward to result from transformations in each transformation and decide answer with the highest similarity and reward.

8. References:

Goel, Ashok. Joyner, David. Thaker, Bhavin. KBAI eBook: *Knowledge-based Artificial Intelligence October 6,2016* [Retrieved From] https://files.t-square.gatech.edu/access/content/group/gtc-bf70-dcdf-5e4b-a676-3e2fb8df4997/KBAI-Ebook/kbai_ebook.pdf

Kunda, Maithilee. McGreggor, Keith. Goel, Ashok. June 2013 A computational model for solving problems from the Raven's Progressive Matrices intelligence test using iconic visual representations [link]

Hunt, 1974 E. Hunt**Quote the raven? Nevermore!** L.W. Gregg (Ed.), Knowledge and cognition, Erlbaum, Hillsdale, NJ (1974), pp. 129-158

Hsu Ming June 2015 [link] http://neuroscience.berkeley.edu/decisions-in-the-brain/

Goel, Ashok. 2017 Basic Problem Slides [link] https://docs.google.com/presentation/d/1Ui57DmPwD0h1pwBd8g1jF8gszpgGHWHpU6n_W99kSRI/edit#slide=id.p13 [Problems] *1 to 24*

Goel, Ashok. Joyner, David. Udacity Video Course: *Knowledge-Based AI: Cognitive Systems* [Course link] https://classroom.udacity.com/courses/ud409 [Lessons] *1,2,3,4,5,6,7,8*

J Graph Ltd. *2005-2017* [link] https://www.draw.io/