Traci Fairchild
CS7637 KBAI
Assignment 1
August 29, 2015

Addressing Raven's Progressive Matrices

With this assignment, I will be describing how I plan to write an *AI agent*, in Python, to solve one or more of the Raven's Progressive Matrices problems (RPM's). My goal with this attempt is to write a series of classes that will process information similar to the way humans think and in doing so, come up with comparable results given a set of RPM's. I am going to address how I would go about doing this using the Semantic Networks design that I've recently learned about in class. In this way I can further examine what intelligence is, and if by mapping this intelligence to a computer generated process then I might be able to understand and explore the relationship of human thought to computer processes. When this field is thoroughly explored, and when human and computational potential finally converge, can we call humans and computers equals as far as intelligence is concerned? Will the two entities ever completely converge? Of course, this assignment will barely scratch the surface of a long debated topic, but surely it will lay the groundwork to my understanding of the functionality necessary to further explore the field of AI.

What makes this problem difficult is that we have to first understand the visual cues that a human would normally notice and then find a way to use this information within the agent itself. The challenge becomes setting up a logical framework of data structures and functions to process information intelligently.

This solution will be designed using semantic networks. In understanding the relationships between the figures in an object one can leverage the simplicity of designing an agent in this way. I will begin by using the provided problem set information of a figure which has been stored in an instance of the RavensObject class. I will take image A and image B of the problem set and loop through each attribute, comparing them against each other. I will also have image C on hand so that, when I compare each attribute in A and B, then I can accurately "build" the desired answer for an "if A is to B, then C is to ?" problem. Let me illustrate with an example.

Image A - 2 figures	Image B - 1 figure	
figure a	figure c	Difference
shape: heart	shape: heart	unchanged
size: very large	size: very large	unchanged
fill: no	fill: no	unchanged
figure b	figure n/a	
shape: diamond	n/a	removed
size: very small	n/a	removed
fill: yes	n/a	removed
inside: a	n/a	removed

Now I know the relationships between A and B, so I can apply the same ones to C and come up with a mapping of what our solution should look like. I also note that there should only be 1 figure in the solution. See as follows:

Image C	Solution?
figure e	conclusion - should have 1 figure only, with the following information:
shape: octagon	shape: octagon
size: very large	size: very large
fill: no	fill: no

With this map of what the solution should look like, my agent will then read through each answer looking for the specifications determined by our comparison above. This example was taken from Basic Problem B-11 and the correct answer would be 1 - which is what the agent would pick out from the answers given, based on the conclusion above.

As I process each figure, I will save it and it's information into a knowledge database for future use. I will store the actual image or it's bitmap representation, along with their attributes so I can use this in later projects to identify shapes in more complicated images. If processing costs remain low I will then manipulate each image - such as flip, rotate, fill, unfill, etc. saving this new image for the knowledge base as well.

What makes this problem difficult is that I am relying completely on the given textual representation of each image, as provided in ProblemData.txt. The challenge problems do not have this so I must come up with an additional solution. I suspect this will be in the form of using the knowledge database that I generate which may contain some or all of the pictures given in the challenge problems, manipulated in some way. I also feel that for the challenge problems it might be a matter of identifying easy relationships across and down the comparison matrix, such as if a = b = c, and d = e = f, then g should = h should = i. This would give the agent good insight to picking the correct answer.

Another problem with using semantic networks is that comparing figure a to figure c for example might not be the right way. Maybe it would make more sense to compare figure a to figure d? This is another major problem if the textual representation is not given.

I believe my methodology will work well for project 1 as long as I can build my object comparison matrices and my analyze the attributes consistently across pictures. Good logic is necessary in comparing and drawing conclusions.