Scott Dickerson
rdickerson3@gatech.edu

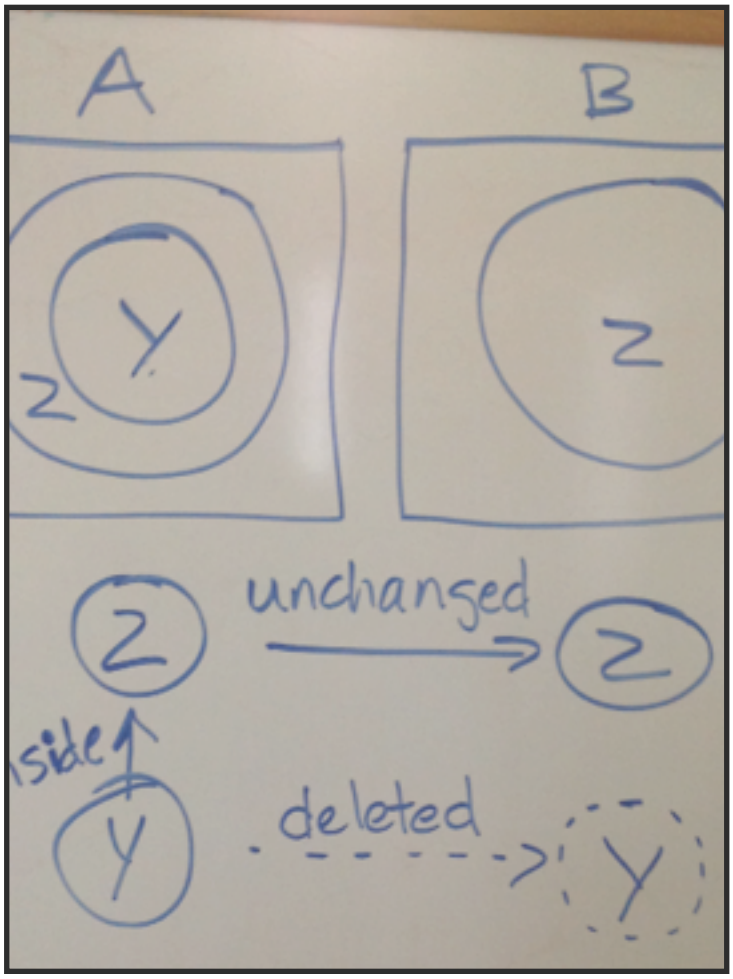
How would you use Semantic Networks to design an agent that can answer Raven's progressive matrices?

A Semantic Network is a way of representing knowledge.

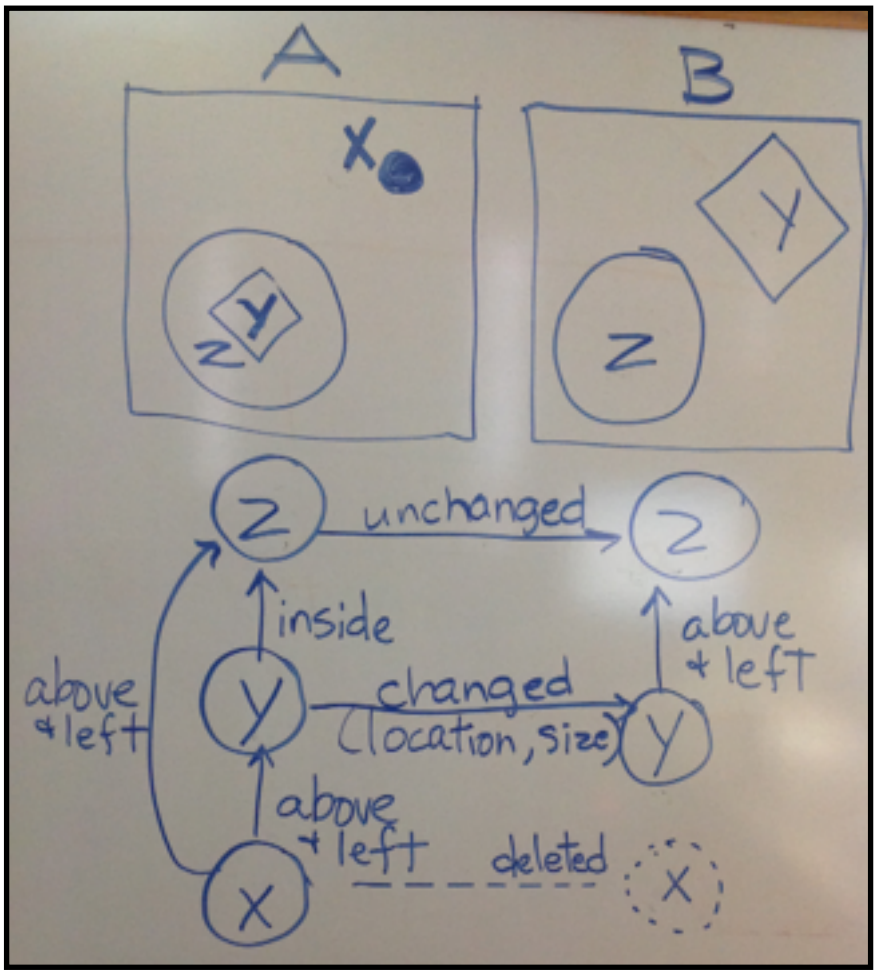
So what do we know about the problems in Raven's progressive matrices?

- We know that each problem contains a set of figures that show a transformation from one to the other(s) and an incomplete set of figures where the final figure of the transformation is unknown.
- Each figure contains one or more objects (or shapes).
- Objects can be just an outline or be filled.
- An object has a natural base.
  - Any side of square or right triangle would be it's base.
  - Any point on a circle would be it's base.
- An object could have been rotated some number of degrees.
  - A square rotated 45 degrees would look like a diamond.
  - A right triangle pointing left would have been rotated 90 degrees.
  - A right triangle pointing down would have been rotated 180 degrees.
  - A right triangle pointing right would have been rotated 270 degrees.
  - A circle rotated any number of degrees would appear the same.
- An object could be incomplete, missing one or more quadrants of the object.
  - Quadrants are numbered 1, 2, 3 and 4 starting with top right moving counter-clockwise.
- The objects in a figure are positioned relative to each other.
  - An object could be inside, above, below or to the left or right of another object.
  - Objects could overlap another object.
- Objects have a relative size compared to the container itself.
  - An object that fills the container would be large.
  - An object that fills only half the container would be medium.
  - An object that fills only a quarter or less of the container would be small.

- Each object in a figure needs to be labeled. Let's start labeling objects as they appear in the figure from the bottom to top, from left to right and, if objects overlap label from front to back. Assign objects letters starting with Z and continue backwards through the alphabet.



The above diagram shows one set of figures (A and B) from a problem. It shows the relationship between the objects in Figure A, characteristics of object Z and Y and the transformation from A to B where object Z did not change and object Y has been deleted.



The diagram above shows a slightly more complicated set of figures where the transformation includes object Y moving and becoming larger and object X disappearing.

- We know some things about the transformation that occurs between one figure and the next in a set. For instance, during the transformation an object may be:
  - unchanged
  - mirror image
  - rotated, plus we know how much it rotated (e.g. 60 degrees)
  - scaled, plus we know if it is larger or smaller
  - changed
  - deleted
  - a combination of one or more of the above.

How do we store what we know about these problems in a Semantic Network?

First, let's define the concepts that exists in our knowledge base and what information we what want to store with each concept.

Problem
- id (number)
- type (2x1,2x2,3x3)
- figures
- transformations

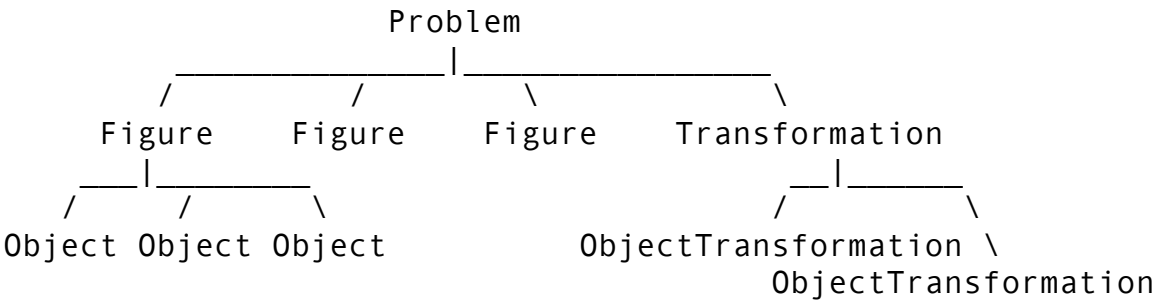
Figure
- id (uppercase letter or number)
- objects

Object
- id (lowercase letter)
- shape
- size (large, medium, small)
- filled?
- rotatedBy (45,90,180,270)
- missingQuadrants (1,2,3,4)
- relationship to another object (above, below, left, right, inFront, behind)

Transformation
- fromFigureId
- toFigureId
- objectTransformations

ObjectTransformation
- objectId
- how
- howMuch (degrees, scaling factor)

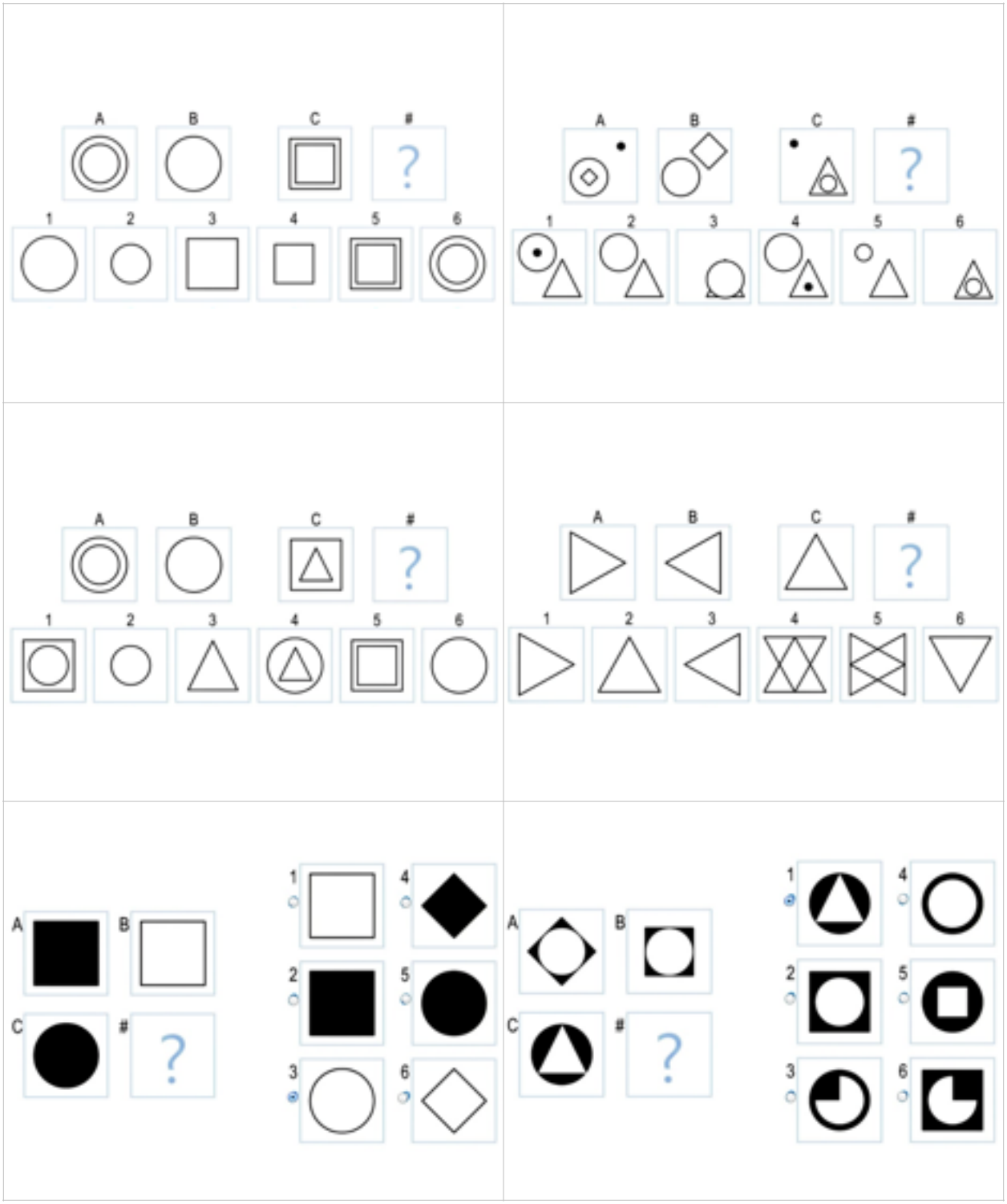
Here’s a graphical representation of these concepts and how they are connected:

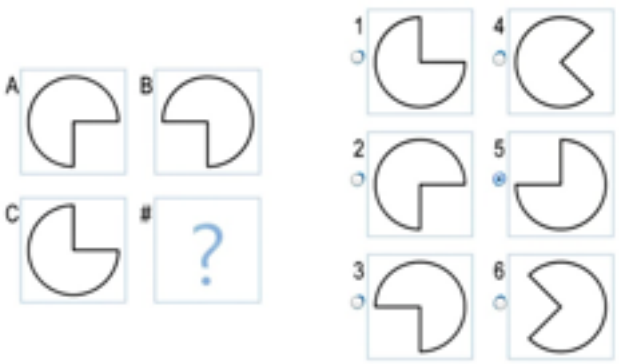


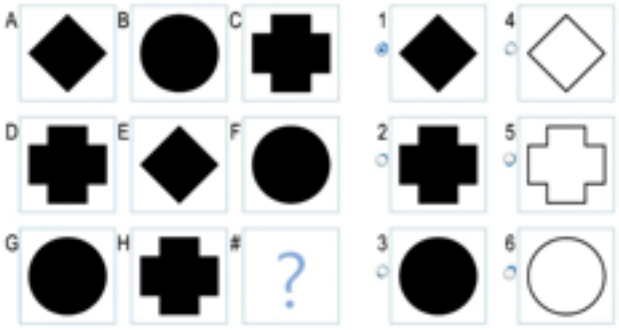
The diagram shows how a problem consists of multiple figures which consist of multiple objects. A problem also consists of one or more transformations which consists of multiple object transformations.

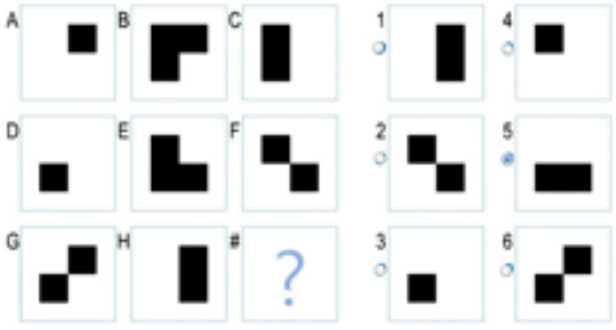
To build out the Semantic Network analyze each problem, extracting the types of knowledge identified above and store in a text file in a format that is easily be read by an AI agent. The AI agent will be responsible for extracting this knowledge and organizing it in memory. The AI agent will handle determining what transformations have occurred and storing those in memory. The AI agent will utilize the knowledge stored in the Semantic Network to solve the problems in Raven’s Progress Matrices by applying a problem solving method such as Generate and Test.

Sample problems from Raven's Progressive Matrices




A
B
C
#
?
1
2
3
4
5
6


A
B
C
D
E
F
G
H
#
?
1
2
3
4
5
6


A
B
C
D
E
F
G
H
#
?
1
2
3
4
5
6