Robert Johns
CS7637: Knowledge-Based AI
Assignment 1
January 19, 2018

Representing Raven's Progressive Matrices Using Semantic Networks

I. Introduction

In solving any given puzzle or problem, representation of knowledge is key. In *Artificial Intelligence*, Patrick Winston states that "once a problem is described using an appropriate representation, the problem is almost solved" (Winston, 1992, p. 18). This rings especially true when writing a computer program to solve a given puzzle or problem. Whereas a human being can push all representations and thought processes into their subconscious, remaining unaware of the methods they're actually employing, a computer lacks such facility, and needs to be explicitly programmed to complete a task; to accomplish this, the structure and content of the problem needs to be clear in the program.

II. Semantic Networks

Semantic networks, a type of graph that represents semantic relationships (i.e. those with a specific meaning) between objects, form a strong basis for knowledge representations in solving Raven's Progressive Matrices (RPMs), a type of geometric analogy problem. Both semantic networks and RPMs are fundamentally discrete structures; semantic networks represent a finite number of qualitative relationships between a finite number of objects, and RPMs are problems with a finite set of choices between images that contain a finite set of figures related in ways that are easy to represent semantically. After building descriptions of individual figures and describing how figures are related in a single image, we can then catalog changes from image to image. After these two steps are executed, we will have built a semantic network for the problem, and can then employ one of several problem-solving methodologies (e.g. generate-and-test, or describe-and-match) using this representation.

III. Structure

To understand the structures of the semantic networks we will build, it's important to understand what it is we're representing. An example RPM problem is shown below:

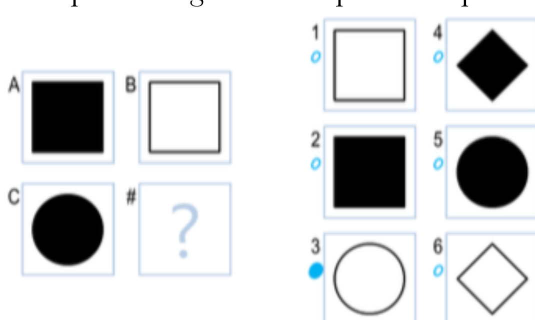

Figure 1: Example RPM Problem (Goel, Joyner & Thaker, 2016, p. 23)

Our semantic network will represent the following structures:

- **Figure**: a figure is a Euclidean shape (in the example above, we have squares and circles), each given an ID. Figures will have the following properties:
    - Shape: a geometrical description of the figure (e.g. square, Pac-Man, plus)
    - Angle: the angle, relative to a defined standard position, that the figure has been rotated
    - Fill: whether or not the figure's interior is whitespace or blackspace
    - Size: the figure's size, relative to a defined standard
- **Image**: an image is a collection of figures, each named (in the example above, we have A, B, C and 1-6). The spatial relationships between the figures in an image are critical when constructing the semantic network and comparing potential solutions. Our agent will describe the following relationships between figures in a given image:
    - Overlaps: a set of figures which our figure overlaps.
    - Inside: a set of figures of which our figure is inside.
    - Above: a set of figures of which our figure is above
    - Left-of: a set of figures of which our figure is to the left.
- **Problem**: a problem is a collection of images, with three or eight being "clues", and 1-6 being choices, of which one is correct. It is the goal of our agent to choose the correct image.

In the example problem given, the high-level goal is to answer the two analogy questions "A is to B as C is to what?" and "A is to C as B is to what?" with the same image from 1-6. As such, we will also need to describe the relationships *across* images. We will describe the following transformations:

- Expand: if the figure increases in size
- Shrink: if the figure decreases in size
- Fill/unfill: if the figure becomes filled or unfilled
- Rotate: if the figure rotates
- Reflect: if the figure is reflected across a horizontal, vertical or diagonal axis

Our final result will look something like this:

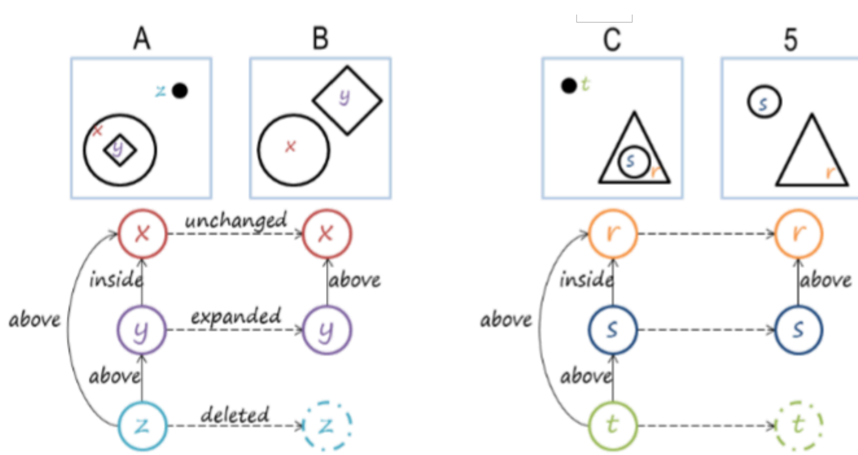

Figure 2: Example Semantic Network for an RPM problem. (Goel, Joyner & Thaker, 2016, p. 30)

With the figure descriptions, spatial relationships and transformations catalogued, our program will have a clear representation of the problem. This representation can then be employed with relative ease into a problem-solving method. For example, we could use a generate-and-test method: once the transformations T1 (from A to B) and T2 (from A to C) have been described, use T1 to transform C into a possible answer, and use T2 to transform B into a possible answer (potentially merging the two into a single image). Then, find the image 1-6 that best matches the generated answer(s). We could also employ a describe-and-match problem-solving method: once T1 and T2 have been described, describe the transformations from B and C to each of the options 1-6, then choose the option whose transformations "best match" T1 and T2.

The point of describing both of these problem-solving methods, despite the fact that neither is the subject of this paper, is to show how flexible semantic networks are as a representation scheme: once built, they can be employed by a diverse range of problem-solving methods. Also, the implementing of the given problem-solving process becomes very straightforward with this representation scheme, given how clear and accessible the relevant information is in the problem. This clarity and flexibility strongly supports Winston's claim that "once a problem is described using an appropriate representation, the problem is almost solved" (Winston, 1992, p. 18).

IV. Implementation Issues

The main issue that comes up with a semantic networks representation scheme is ambiguity in a figure or transformation. For instance, consider a problem in which figures A and B are both unfilled, large circles in the center of the image. Several possibilities arise: the figures could be unchanged, or they are rotated or reflected in any number of ways on their way from A to B. It is impossible to determine which transformation is correct, and thus what the correct representation in the semantic network is, without the context of examining the six possible answers.

Related to ambiguity is another problem, that of determining which figure corresponds to which across images. Take the image pair below, for example:

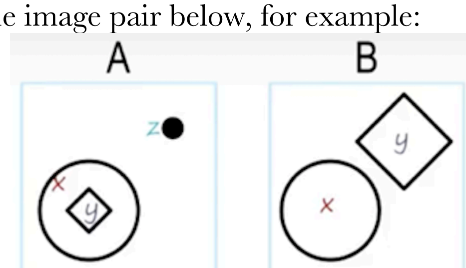

Figure 3: Example Pair of Images (Goel, Joyner & Thaker, 2016, p. 31)

Without comparing all possibilities, it's impossible to determine if, in the transition from A to B, dot *z* disappeared and square *y* expanded and moved from inside the circle to the upper left, or if square *y* disappeared and dot *z* was transformed into a large, rotated square. The ambiguity in determining the transitions between figures is impossible to resolve without considering the larger context of the problem, and even then, it can be difficult.

Whatever problem-solving method the agent employs will need to account for these various types of ambiguities, either by employing weighted similarity metrics to determine the most likely of the possibilities, or by considering each one in the problem-solving method (and thus risking combinatorial explosion), the use of semantic networks allows for great flexibility.

V. Conclusion

Semantic networks more than meet the principles of good knowledge representation outlined in Winston's *Artificial Intelligence* in the problem domain of Raven's Progressive Matrices. The relationships between our figures, both within and across images, are made explicit and clear, and the objects and relationships are brought together in the same network graph. We are able to suppress insignificant detail by describing transformations in a sufficiently abstract manner, leaving out precise quantities if necessary. Semantic networks are transparent, complete, concise, fast, and, as we have seen, are easily computable. Once implemented in our agent, semantic networks should form a strong basis for computationally solving Raven's Progressive Matrices.

References

Winston, Patrick Henry (1992). *Artificial Intelligence, 3rd Ed.* London, UK. Pearson

Goel, Ashok; Joyner, David; Thaker, Bhavin (2016). *KBAI Ebook: Knowledge-Based Artificial Intelligence*. Publisher, city and state not given.