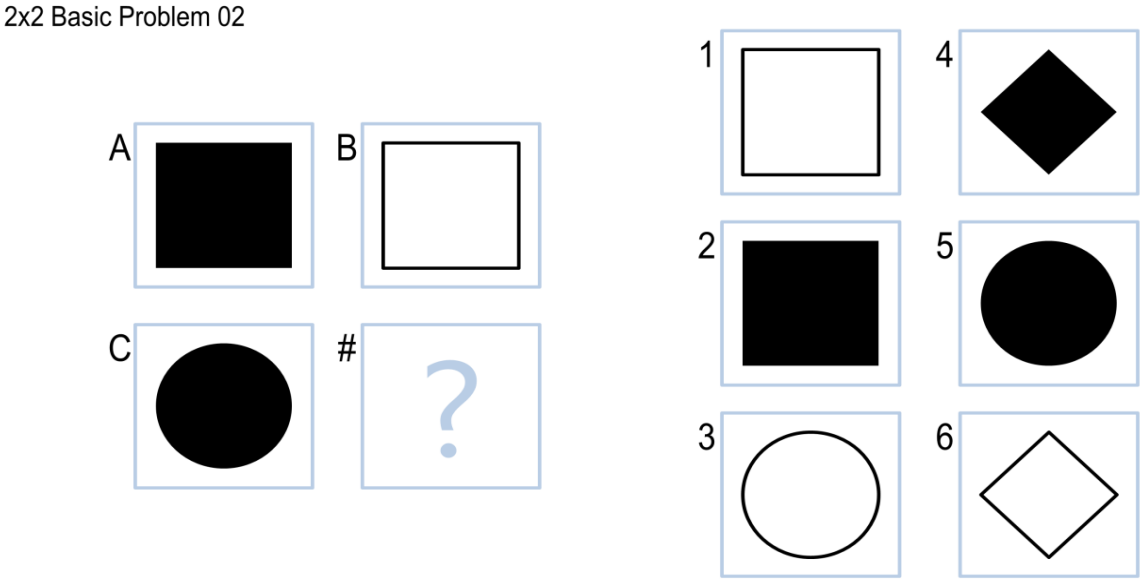
Brad Ware
bware8@gatech.edu
Assignment 4

Using Means-End Analysis to solve Raven's progressive matrices

When typically approaching a problem, I am always concerned with how to find the "right" answer, and usually design my approach around that. This seems like a very natural tendency, but may not be the best path to the answer. Many of the problems on the SAT can be solved by simply eliminating the answers that are wrong, which reduces the state space of all possible answers to a manageable size. This strategy is extremely easy on multiple-choice problems; the answer state space is already reduced! For this assignment, I want to talk about using heuristics and information about the goal to eliminate possible answers for Raven's progressive matrices. This approach is similar to problem reduction, but because the elimination is due to inferred knowledge on the goal, it falls more under the category of Means-End Analysis.

Means-End analysis is a problem solving strategy that analyzes the goal state and tries to identify procedures or steps that constantly allow the current state to become closer to the goal state, until it is reached. In the problem of Raven's progressive matrices, the goal state is one of the answer figures that most closely associate the matching of figures AC and AB. Because this problem is not as straightforward as a search, one has to be careful on how they define becoming closer to the goal state. The key attributes here are differences in images, as that pertains to everything in this situation for knowledge inference. Let's take a look at this Raven's problem for example:



In terms of Means-End analysis, the Agent would first evaluate the goal state. In order to determine what the goal state should be, it will evaluate the pairings of AB and AC, and dump the knowledge differences into a storage space.

AB Space = {“fill”}
AC Space = {“shape”}
Problem2Space = {“fill”, “shape”}

After we have some inference about our goal, our agent can then loop through each answer choice, and eliminate answers that don’t bring our agent closer to the goal.

```
For currFigure in Problem2.getFigures():

        AFigSpace = analyze(A, currFigure)
        CFigSpace = analyze(C, currFigure)
        totalSpace = comine(AFigSpace, CFigSpace)

        for element in Problem2Space:
                if(totalSpace does not contain(element))
                        eliminate(currFigure)
```

So through this simple pseudo code, our agent is eliminating the possible answer choices that contain differences that have no correlation to the goal state. This is exactly what Means-End analysis provides, steps that only lead closer to the goal. So if we applied this pseudo code to Problem 2 above:

Figure 1 – would be kept as possible answer choice because it’s space would be the exact same Problem2Space - shape and fill differences.

Figure 2 – this figure would be eliminated because it does not contain a fill difference.

Figure 3 – would be kept because it follows the same procedures to become closer to the goal state

Figure 4 – would be eliminated because it contains an unnecessary trait of angle change, which does not bring us any closer to the goal state. It also has no fill change.

Figure 5 – Would be eliminated as well, no fill change, which is necessary to reach goal state.

Figure 6 – Unnecessary angle change, does not help our agent reach the goal state

Obviously, this pseudo code was very high-level, and not all of these images would be eliminated in the same round. For example, only 2 and 5 would be eliminated the first pass through, because they did not contain the bare minimum of change to reach the goal state. After closer analysis the agent would then prefer to keep the figures that replicate the same amount of change, and not any unnecessary transitions. Therefore, our agent would then eliminate both of the figures that had angle changes within their shape, 4 and 6. Finally, our agent is only left to decide the answer with two possible figure choices, 1 and 3! It now has a 50% chance of getting the correct solution by random guess, where it started with only a 17% possibility. Obviously with a slight comparison it would now check to see that the figure's shape should be different from B because of the AC change space, and choose 3 as the answer. Means-End analysis did not provide the correct solution in this space, but it did reduce the problem's possible answer space to a much more manageable size. This alone gives the agent a higher probability of choosing the right answer.