

Project 2: Raven's Progressive Matrices

Marc Micatka

mmicatka3@gatech.edu

1 INTRODUCTION

The agent developed to solve Project 1 will be updated and revised to apply a similar, purely visual, problem-solving approach to 3x3 RPMs.

In Project 1, the problems are 2x2 and the transformation is often basic – a simple geometric or affine transformation of the original image. In Project 2, the problems are 3x3 and the transformations are more complex. **Figure 1** shows a basic problem from problem set C with a simple transformation (outer square growth from left to right) but it does not fit easily into an affine transformation (rotation, mirroring, identity).

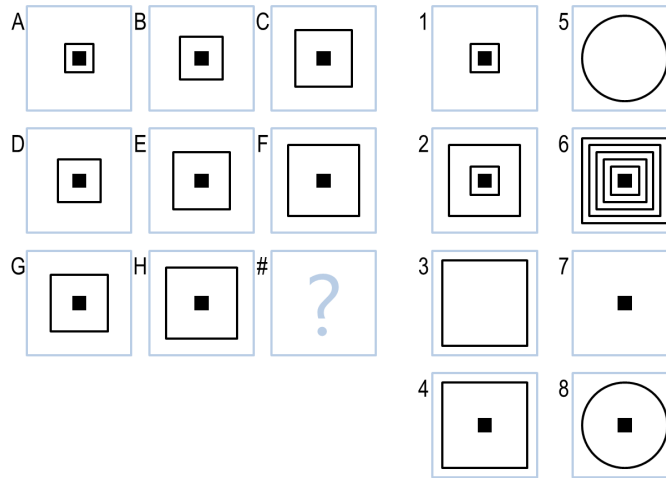


Figure 1 – Basic Problem C-2 demonstrates the more subtle nature of the transformations in Project 2.

The final agent designed for Project 2 attempts to follow the same approach as the agent designed in Project 1. First, different affine transformations are calculated for horizontal and vertical figures. This includes direct transformations like A to B but also indirect, like A to C. The direct and indirect transformations are computed for D to F, E to F, E to H, and B to H. The transformation (the different transformations are shown in Appendix 4.1) that returns the highest similarity score is chosen along with the direction that resulted in that score (direct versus indirect, horizontal or vertical).

The agent then calculates the similarity between the H/F or C/G and choices 1- 6 using only the transformation chosen in earlier steps. The option that returns the highest similarity match over a threshold value is chosen as the answer. If no answer scores over the threshold, the agent will revert to simpler weighted voting method applied in Project 1.

Basic Problem C-07 (shown in Appendix 4.5 and in Figure 2) will be used as an example. Initially, A will be compared to B and will return the highest similarity value and the transformation that returns this score. This is repeated for the other comparison images. For C-07, “Horizontal, Indirect”, returns the highest score for “Y-Axis Mirror” across all transformations and all images. The agent would then *only* assess image G and the answers 1-6 for this transformation.

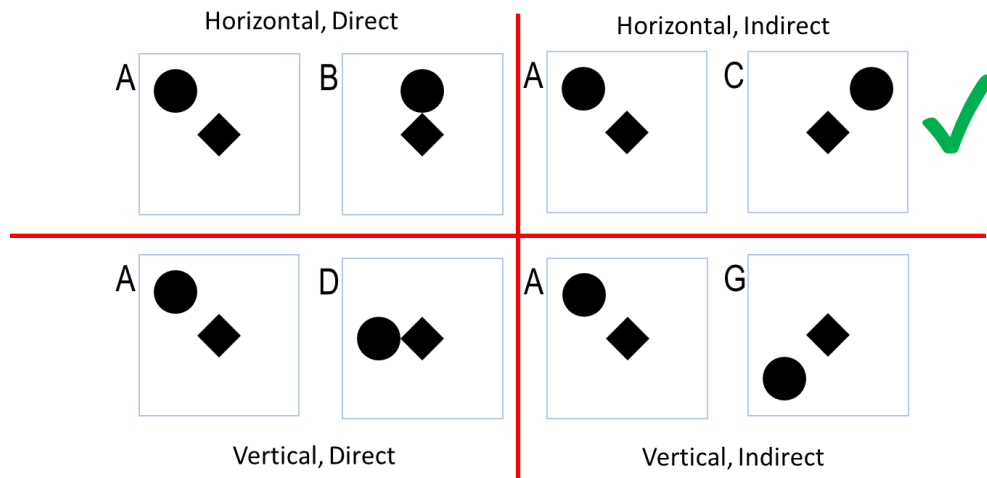


Figure 2 — Problem solving approach to Basic Problem C-07.

2 JOURNAL ENTRIES

2.1 Submission 1: 2019-10-17 22:28:35 UTC

Problem Set	Correct	Incorrect	Results
Basic Problems C	5	7	42%
Test Problems C	6	6	50%
Challenge Problems C	2	10	17%
Raven's Problems C	5	7	42%
Runtime	-	-	10.361 s

What did you change for this version? Why?

This was the first submission and I wanted a performance baseline for my revision. For this submission, I reused my Project 1 code and adjusted my comparison images, from A-B to E-F and from A-C to E-H, treating the 3x3 matrix like a 2x2 matrix confined to the lower right corner.

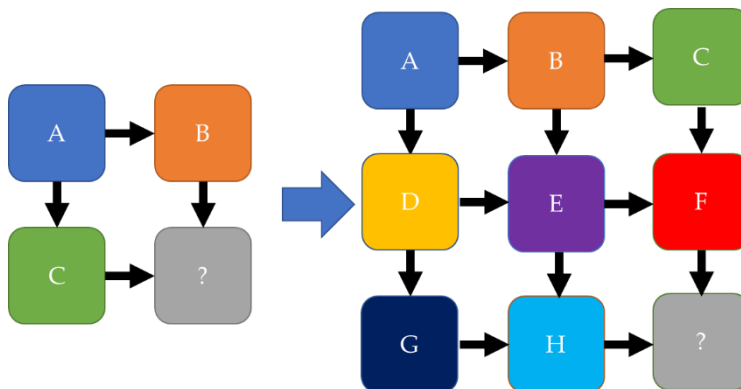


Figure 3— 2x2 matrix transforms as compared to 3x3. This is for reference when referring to different selections.

How would you compare this version of the agent to the way you feel you, a human, approach the problems? Does it “think” similarly to how you think, or differently?

In the most basic sense, this algorithm does attempt to mirror human cognition. It looks for similarities to determine the transformation T and then finds the best match among the choices. The weight matrix to determine the solution is fundamentally similar to human cognition but differs significantly in operation. While humans do weight certain transformations higher than others (like identity or rotation over pixel ratio) it would be difficult to generate a matrix of weights that would determine the cutoff point. In addition, I did quite a bit of hyper-parameter tuning to dial in the performance for Project 1. This meant that some weights, like rotation, were lower than mirroring. This doesn't make logical sense, but it resulted in better performance.

The biggest cognition difference is that this agent only considers 3 images to generate a guess, not the full set available.

How did it perform? What problems or types of problems did it do well on? Where did it struggle? How is its efficiency?

This version did okay – clearly room for improvement as I didn’t hit the threshold for either the Test set or the Basic set and I struggled on the Challenge set. There are many problems in Set C that require full knowledge of the 3x3 matrix and on these images, the agent struggled. It is also relying heavily on the pixel ratio metric (and will for many submissions going forward). This metric can be a mixed bag because it can give a perfect match without having any knowledge of translation or rotation. For instance, a triangle that is mirrored will return a perfect pixel ratio metric and will result in the agent choosing another triangle, regardless of orientation. This solution is very efficient, both from a runtime perspective (10 seconds to solve 96 problems) and from a mathematical perspective. Each answer is evaluated only once and only 4 of the 8 provided images are assessed.

2.2 Submission 2: 2019-10-19 20:58:37 UTC

Problem Set	Correct	Incorrect	Results	
<i>Basic Problems C</i>	6	6	50%	▲ +1
<i>Test Problems C</i>	5	7	42%	▶ --
<i>Challenge Problems C</i>	3	9	25%	▲ +1
<i>Raven’s Problems C</i>	8	4	66%	▲ +3
<i>Runtime</i>	-	-	9.015 s	▲

What did you change for this version? Why?

The first submission was based on my Project 1 submission – it chose the answer based on a holistic evaluation of all the transformations (see Appendix 4.1) without direct consideration of the single best transform. This works well for simple transformations but fails when there is one single transform that describes the analogy. Basic Problem C-05 (see Appendix 4.2) is a good example of this issue. For the second submission, I updated the selection algorithm to first search for the single best transform that describes the transformation between E and F and between E and H (horizontal and vertical). The agent then searches the answers for a match above a confidence threshold and, if found, selects that answer. If no answer exceeds the threshold, the agent reverts to the weighted matrix method from Project 1.

How would you compare this version of the agent to the way you feel you, a human, approach the problems? Does it “think” similarly to how you think, or differently?

Again, this version only considers a small 2x2 matrix subset of the 3x3 problem which is not human-like. The first search from the agent is more like human cognition than the weighted matrix search. Humans will scan the problem and try to find a transformation that best describes what is happening between the images. Although we won’t assign a “confidence” score like the agent, we do tend to rank our answers internally based on how confident we are in the solution.

How did it perform? What problems or types of problems did it do well on? Where did it struggle? How is its efficiency?

This version improved on submission 1 on all but the test set. It does a good job of evaluating whether the transformation is horizontal or vertical and performs well on simple affine transformations and transformations that involve adding objects in a geometric sequence (see Basic Problem C-03 in Appendix 4.3). Even though the agent has no knowledge of objects, shapes, or addition, it can see that dark objects are being added in a set ratio that should be maintained in the answer. This solution fails, and the agent struggles, when more complex reasoning is required to understand changing white/dark pixel ratios (see Basic Problem C-12 in Appendix 4.4). For this problem, the agent would need more complicated logic to understand which white squares are disappearing through the transformation.

Adding in the first search method improves the efficiency of this algorithm because certain logical assessments can be avoided. When the agent is able to find a solution with the first method, it does not need to calculate the affine transformation matrix which saves time and reduces computation load. This can be seen in the shortened runtime.

2.3 Submission 3 and 4: 2019-10-19 21:03:18 UTC

Problem Set	Correct	Incorrect	Results	
Basic Problems C	8	4	66%	▲ +2
Test Problems C	6	6	50%	▲ +1

Problem Set	Correct	Incorrect	Results	
Challenge Problems C	3	9	25%	► --
Raven's Problems C	9	3	75%	▲ +1
Runtime	-	-	10.497 s	▲

What did you change for this version? Why?

Because I misread my submission data in Bonnie, I accidentally submitted the same version twice here. For this submission, I changed my error metric from a basic similarity check on Euclidean distance to a mean square error check which returns much more accurate results when images are close but not exact. This helps the agent solve some problems that register as different using the Euclidean metric. Because of small errors in PIL rotation and mirroring algorithms, transformations that appear to be perfect affine transformations register as imperfect. The improved accuracy in the metric also solves images with more difficult transformations like Basic Problem C-05 (Appendix 4.2).

How would you compare this version of the agent to the way you feel you, a human, approach the problems? Does it “think” similarly to how you think, or differently?

This agent improves its human-like cognition by using an error metric that considers objects that are “close” to the original as more similar than the Euclidean metric would. Humans are exceptionally good at looking at shapes and understanding similarity whereas AI agents must do some calculation to arrive at that same understanding. The MSE metric tries to improve on this shortcoming. Apart from the metric change, there were no fundamental changes to the cognition of the agent.

How did it perform? What problems or types of problems did it do well on? Where did it struggle? How is its efficiency?

This version does well on the Basic Problems but failed to improve over baseline on the Test set or the Challenge Set. Although it’s unclear why the agent cannot improve on the Test set, the Challenge set includes a lot of 2-part transformations that will require a better understanding of the changes occurring in the 3x3 matrix instead of restricting the performance to the 2x2 corner. The efficiency has decreased a bit because MSE is slightly more computationally intense to calculate.

2.4 Submission 5/6: 2019-10-23 11:27:48 UTC

Problem Set	Correct	Incorrect	Results
Basic Problems C	8	4	66% --
Test Problems C	6	6	50% --
Challenge Problems C	5	7	42% +2
Raven's Problems C	7	5	58% -2
Runtime	-	-	10.497 s

What did you change for this version? Why?

Once again, I accidentally submitted the wrong version initially so I'm only discussing the more correct submission for this version. For this update, I examined the challenge problems and noticed many have complicated transforms between A-B or B-C but the transform between A and C is much simpler. Challenge Problem C-07 is a good example of this (see Appendix 4.5).

To see if I could move the needle on the Test and Challenge problem results, I added a check for D-F and B-H. This improved my performance on the Challenge Problems but did not affect my performance on the test data set.

How would you compare this version of the agent to the way you feel you, a human, approach the problems? Does it "think" similarly to how you think, or differently?

This agent begins to look at transforms across all 3 images, horizontal or vertical. Although I'm not doing the best job of making full use of the images provided, I'm adding additional information to the agent that a human would consider. I'm still not considering all 8 images when I make my choice.

How did it perform? What problems or types of problems did it do well on? Where did it struggle? How is its efficiency?

This agent performs better on the Challenge problems but sees no performance improvement on the Basic or Test problem set and performs worse on the Raven's set. I've been trying different approaches to improve my score on the Test set because I have yet to get above 6 correct. This is obviously frustrating because I'm one answer away from a perfect score. This submission performs much better on

problems where there is a clear transformation between A-C (or A-G for the vertical case) but a more confusing middle transition. Appendix 4.5 shows an example of this problem. The actual transformation may be best described by the translation of the objects across the image, meeting in the middle, then swapping sides. But just assessing the 1st and 3rd image makes the transformation much clearer for the agent – a simple swap where we expect the pixel ratio to remain the same between the 1st frame and the answer.

2.5 Submission 7: 2019-10-26 10:37:26 UTC

Problem Set	Correct	Incorrect	Results	
<i>Basic Problems C</i>	9	3	75%	▲ +1
<i>Test Problems C</i>	6	6	50%	▶ --
<i>Challenge Problems C</i>	3	9	25%	▼ -2
<i>Raven's Problems C</i>	3	9	25%	▼ -2
<i>Runtime</i>	-	-	9.767 s	▲

What did you change for this version? Why?

In an effort to make my agent perform in a more human-like way, I added an initial check to determine the best possible affine transformation (Appendix 4.1) and where that transformation occurred (horizontal E-F, horizontal D-F, vertical E-H, vertical B-H). The agent then compares the equivalent image (H, G, F, or C respectively) and finds the best match for that specific transform. If none of the transforms score over a given threshold (0.90 MSE similarity) the agent defaults to a weighted average from Project 1. I wanted to add some logic to allow my agent to more accurately identify the correct transformation *direction*, horizontal versus vertical, as well as the correct transformation itself.

I also noticed something a bit peculiar in how the agent was determining matches. My agent was almost always choosing “Pixel Ratio” as the best match even when there was a clear match for identity (like in Basic Problem C-01). This is because of very small calculation errors that would return a 1.0 match for pixel ratio between identical images but a value of 0.9999 for identity. To get around this, I added a small weight factor to my transforms, treating identity as 1.0, mirroring and rotation as 0.95, and pixel ratio as 0.90. This helped choose the correct answer more frequently.

How would you compare this version of the agent to the way you feel you, a human, approach the problems? Does it “think” similarly to how you think, or differently?


I intentionally made some changes to this version to add more human-like logical processing. When humans see an transformation like in Appendix 4.5, we can reason that there are a few transformations going on between A and B (maybe translation, maybe rotation) but the best transformation becomes obvious when we see C. I added that layer of cognition to my agent by calculating which step gives me the best similarity.

How did it perform? What problems or types of problems did it do well on? Where did it struggle? How is its efficiency?

On the basic problems, this agent outperforms all other submissions. I was struggling for a while because between a few different methods, I could capture most of the basic problems, but I was unable to determine how to choose the right method to capture the most possible answers. On the other problem sets, the performance of this agent is much worse or unchanged. This is mostly a reflection of the different types of problems in the different sets – the basic problems are mostly simple affine transformations that don’t require much understanding of shapes, contours, corners...This agent continues to struggle on problems that require more complex reasoning of transformations. Appendix 4.6 shows Basic Problem C-06 which no version of my agent solves. This problem stymies my agent because there is not good affine transformation that describes what’s happening and the change between A-B and B-C in size is not consistent. Even using the pixel ratio metric does not help. My efficiency is quite improved because I no longer must calculate all transformations for all answers, just the one I’ve already identified as my best transformation.

2.6 Submission 8: 2019-10-26 10:39:11 UTC

Problem Set	Correct	Incorrect	Results
Basic Problems C	8	4	66% ▼ -1
Test Problems C	7	5	58% ▲ +1
Challenge Problems C	3	9	25% ► --
Raven’s Problems C	3	9	25% ► --

Problem Set	Correct	Incorrect	Results
Runtime	-	-	8.98s 

What did you change for this version? Why?

I was purely trying to pick up the final test problem to get my 15% credit for this submission. In the previous submission, I had included a threshold of 0.90 for the similarity check. If the check failed, I would default to a weighted average approach described in Project 1. I feel like this is more logical but for this submission I removed that check and used the method described in 2.5 no matter how poorly the answer matched. This resulted in a passing test grade, but low performance outside that result.

How would you compare this version of the agent to the way you feel you, a human, approach the problems? Does it “think” similarly to how you think, or differently?

See the answer in 2.5. Nothing has changed too much from that submission. I think the major decision-making processes are similar to how humans think however it is not able to consider the more complicated transforms like shape detection or translation.

How did it perform? What problems or types of problems did it do well on? Where did it struggle? How is its efficiency?

It performed well where it needed to, on the Test set! Otherwise, I saw a decrease in performance overall as was to be expected. The agent is answering questions that it knows it has a low probability of answering correctly with its chosen method, but it is the only method available (in this version of the agent). Removing the weighted similarity method does improve its efficiency, however, because it has one less check to perform before choosing the answer.

3 CONCLUSION

How would you characterize the overall process of designing your agent? Trial-and-error? Deliberate improvement? Targeting one type of problem at a time?

I began with my framework from Project 1. I knew it had some shortcomings – namely that it could not choose which affine transformation best described the

transformation. The initial changes were very deliberate. As I saw the type of problems it struggled on (skipping the middle) I began to more clearly target specific challenge areas. Basic Problems C-05, 06, and 07 gave me endless trouble trying to find a method that could accurately answer those plus the other, more straightforward problems. Towards the end, I was testing methods to try and get my final Test problem answered correctly because I was sitting at 6 out of 12 for every submission.

How similar do you feel your final agent is to how you, a human, would approach the test? Why or why not?

This is answered exhaustively in every submission note but in general my agent approaches problems like I would from a very high level. It has a bank of simple transforms and it looks at each problem and determines which of those transforms best describe what's happening. Beyond that, humans would be able to look a bit deeper and see more complicated transforms. My most non-humanlike transformation is the pixel count metric which is responsible for a majority of my correct answers. It seems a bit counter-intuitive, but problems like Basic Problem C-03 and C-05 can be answered correctly by looking at the pixel counts between images with no knowledge of which shapes are present or how the shapes or number of shapes is changing.

What improvements would you make if you had more time and/or more computational resources?

With more time, I'd like to add a few layers of understanding to my agent. For Project 3, I think it will be important to analyze the diagonal transformations as well as the horizontal and vertical transformations. I would also like to add blob detection to allow my agent to count connected components and compute the delta between frames. Adding additional metrics like corner detection or simple shape detection would be useful in answering the challenge problems as well.

I would also like to add in some logic layers to help the agent choose when there are multiple transformations present. I added a function to count the number of transformations that scored over the threshold value but I could never use it in a useful manner. I think that humans do this though – we see multiple transformations like rotation and translation – and can apply them both to find the right answer.

4 APPENDICES

4.1 Affine Transformations and Weights from Project 1

Table 1 — Affine transformations used in Project 1 to evaluate answer.

Transformations	Description	Weight
<i>Identity</i>	Compare A and B directly	1.0
<i>Reflection: X</i>	Reflect B across X-axis	0.50
<i>Reflection: Y</i>	Reflect B across Y-Axis	0.50
<i>Rotation: 90°</i>	Rotate B 90° clockwise	0.25
<i>Rotation: 180°</i>	Rotate B 180° clockwise	0.25
<i>Histogram Comparison</i>	Calculate the ratio of white/black pixels	0.25

4.2 Basic Problem C-05

Basic Problem C-05

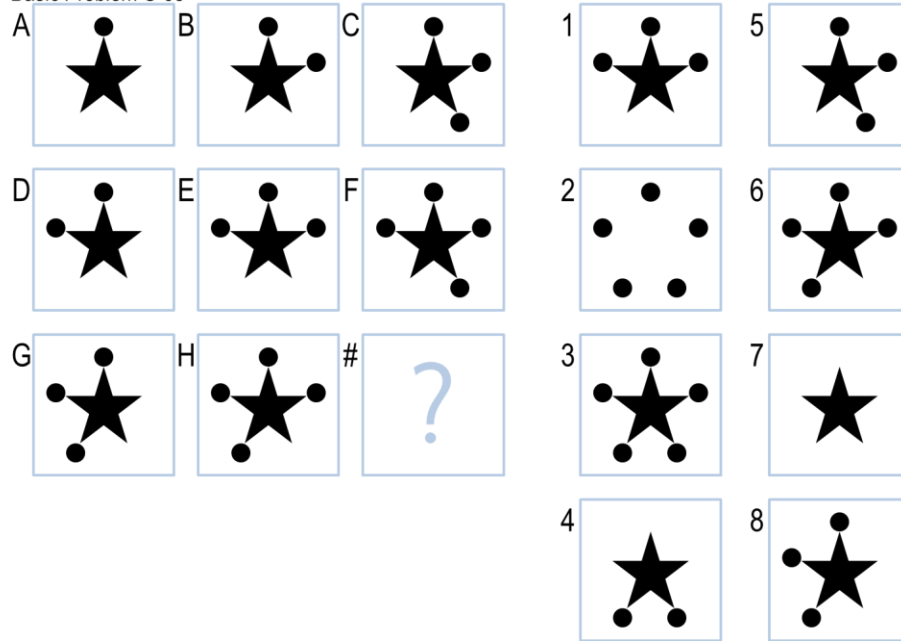


Figure 4— Basic Problem C-05 is a tricky problem to solve with my original solution method outlined in Project 1.

4.3 Basic Problem C-03

Basic Problem C-03

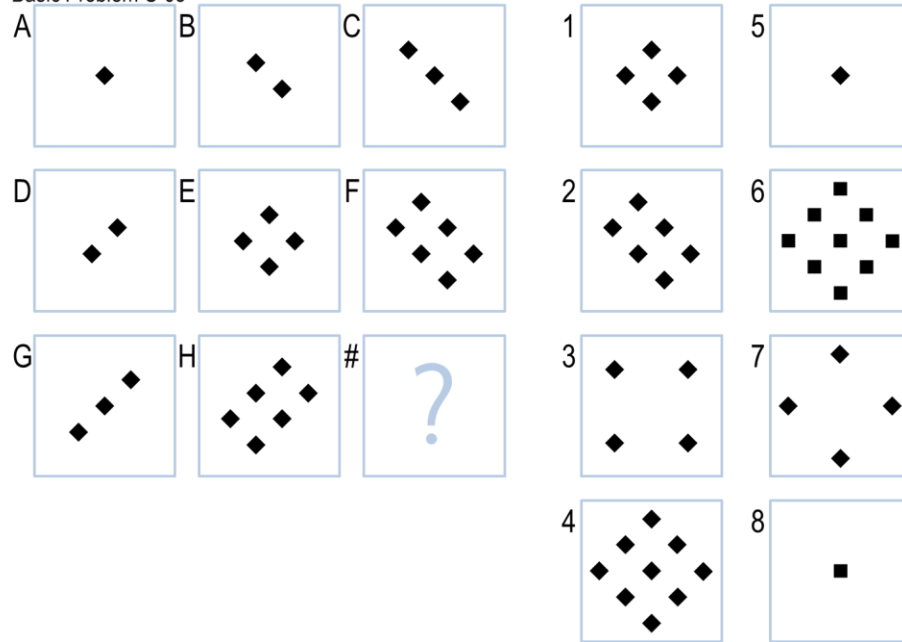


Figure 5— Basic Problem C-03 involves simple addition of dots.

4.4 Basic Problem C-12

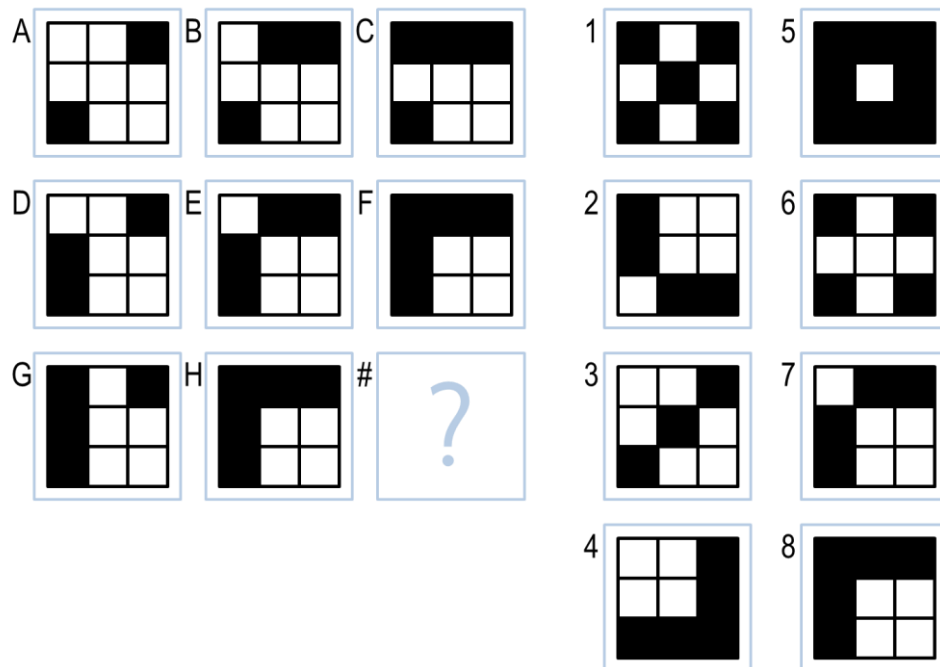


Figure 6— Basic Problem C-12 involves more complicated reasoning to understand which squares are disappearing.

4.5 Basic Problem C-07

Basic Problem C-07

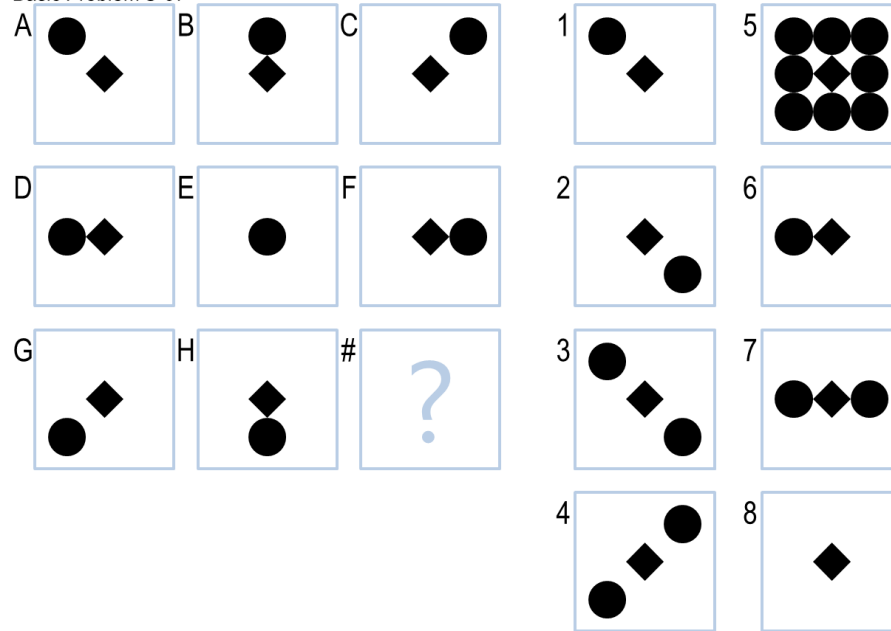


Figure 7— Basic Problem C-07 has a straightforward A-C transformation but a difficult A-B transformation.

4.6 Basic Problem C-06

Basic Problem C-06



Figure 8— Basic Problem C-06 requires a more intelligent agent than my own...