xxiao39@gatech.edu

Project 3- Project Reflection

How does your agent reason over the problems it receives? What is its overall problem-solving process? Did you take any risks in the design of your agent, and did those risks pay off?

In project 2, my agent solved 3x3 matrix problems using lots of verbal information to get attributes of images and a very simple visual method shown below.

difference of Pixels(G->H) $----closest--\rightarrow$ **difference of Pixels(H->?)**

In Project 3, my agent needs to solve 3x3 matrix problems that only have visual representations.
As we know, a cognitive agent is a function for cognitive architectures that maps "percepted" history to "action".

f:P*->A

How to map? My agent first needs figure out some production rules.

Problem D set:
Rule 1: The number of objects has a fixed increment between adjacent images in each column.
Examples: Challenge Problem D-12

Rule 2: total pixel of each column equals one another.
Examples: Basic Problem D-01, Basic Problem D-02, Basic Problem D-03, Basic Problem D-06, Basic Problem D-11, Challenge Problem D-06

Rule 3: if (image C **XOR** image E)==n*image G && (image H **XOR** image A)==n*image F, then image D **XOR** image B->n*answer (n is fixed)
Examples: Basic Problem D-08, Basic Problem D-12

Rule 4: if (image C **OR** image G)==image E && (image H **OR** image F)==image A, then image D **OR** image B->answer
Examples: Basic Problem D-11, Challenge Problem D-09

Rule 5: This is very similar to Rule 4.
if (image E **OR** image G)==image C && (image A **OR** image F)==image H, then image D **OR** image B->answer
Examples: Basic Problem D-09

Rule 6: similar transformations in each row.

A->B->C---similar---D->E->F---similar---G->H->?

Examples: Basic Problem D-04, Basic Problem D-05

Rule 7:if (image B **OR** image D)== image E && image G==image H && image C==image F, then image F **OR** image H->answer.
Examples: Challenge Problem D-01

Rule 8: Not same as image A, image B, image C, image D, image E, image F, image G, image H-> answer
Examples: Basic Problem D-05, Examples: Basic Problem D-07, Basic Problem D-10

Problem E set:

Rule 1: if (image A **XOR** image B)==image C && (image D **XOR** image E)==image F, then image G **XOR** image H ->answer
Examples: Basic Problem E-08, Basic Problem E-07, Basic Problem E-05, Challenge Problem E-01

Rule 2: if (image A **OR** image C)==image B && (image D **OR** image F)==image E, then image G **OR** answer -> image H
Examples: Basic Problem E-06

Rule 3: if (image A **OR** image B)==image C &&(image D **OR** image E)==image F, then (image G **OR** image H)->answer
Examples: Basic Problem E-01, Basic Problem E-02, Basic Problem E-03, Basic Problem E-09

Rule 4: if (image B **OR** image C)==image A &&(image E **OR** image F)==image D, then (image H **OR** answer)-> image G
Examples: Basic Problem E-05

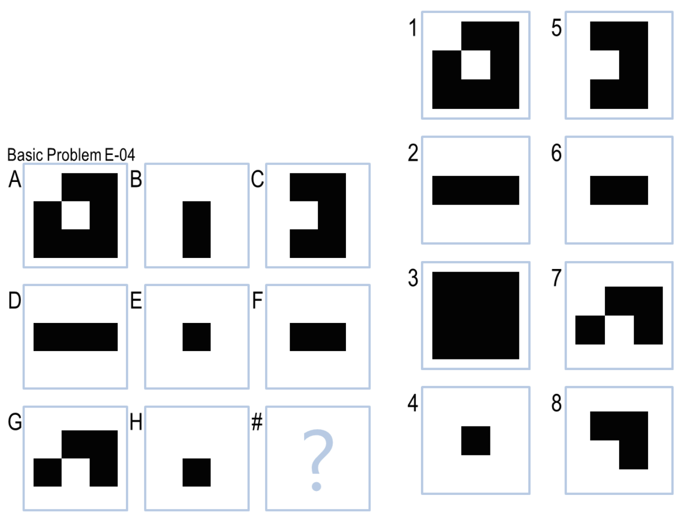
Rule 5: if (image A **AND** image B)==image C && (image D **AND** image E)==image F, then (image G **AND** image H) -> answer
Examples: Basic Problem E-10, Basic Problem E-11

Rule 6: if (top half of image A **OR** bottom half of image B)==image C && (top half of image D **OR** bottom half of image E)==image F, then (top half of image G **OR** bottom half of image H) -> answer
Examples: Basic Problem E-09

Rule 7: if the sum of (image B' pixel and image C' pixel)== image A' pixel && the sum of (image E' pixel and image F' pixel)== image D' pixel, then the sum of (image H' pixel and answer' pixel)==image G' pixel
Examples: Basic Problem E-04, Basic Problem E-12

Next, my agent uses rule-based reasoning, then maps the current problem to production rules and decides on an action.

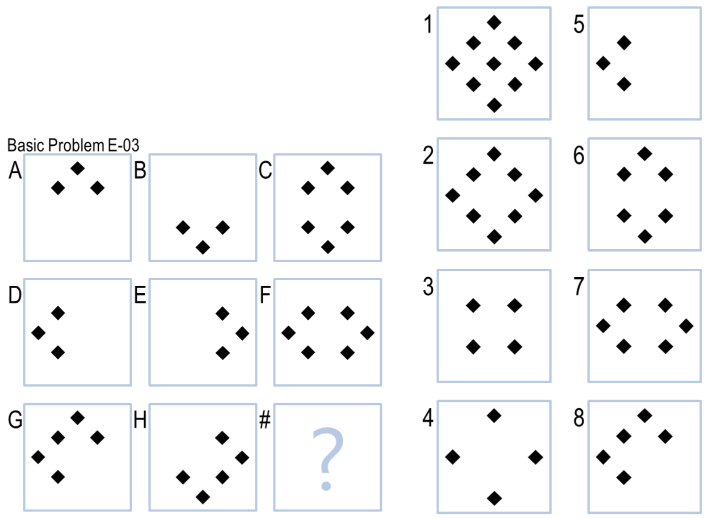
In the design of my agent, my agent cannot figure out a correct answer especially when a problem is mapped to multiple production rules. But my agent has priority of which production rules to decide first.
Depending on production rules, my agent sometimes may not be able to decide on a single operator that is called "impasse". My agent uses an episodic rule to break this deadlock situation.
Here is an example below.



Based on **Rule 7**, my agent gets two operators, image 2 and image 8. My agent construct a new production rule "answer is not same as image A, image B, image C, image D, image E, image F, image G, image H". Finally, the correct answer (image 8) is chosen.

How does your agent actually select an answer to a given problem? What matrics, if any, does it use to evaluate potential answers? Does it select only the correct answer, or does it rate answers on a more continuous scale?

Let us see the problem below.



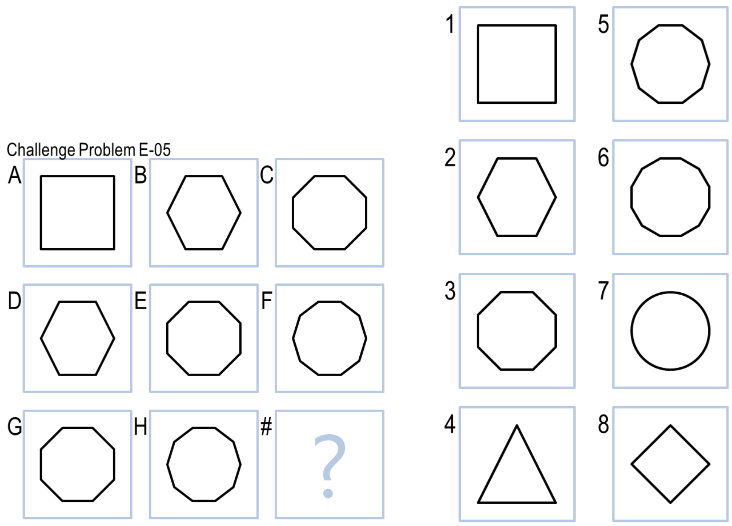
My agent first checks a series of production rules.
Image A **OR** image B generate a new image, then my agent compares image C to this new image to check whether these two images are equal.
Image D **OR** image E generate a new image, then my agent compares image F to this new image to check whether these two images are equal.
Because both are true, this problem matches **rule 3.**
This problem also matches **rule 1.**
My agent gives priority to **rule 3, OR** has precedence over **XOR.** Then my agent reasons (image G **OR** image H)-> answer.
The correct answer (image 2) is chosen.

My agent compares two images by location and color of pixel and gets a matching percentage.
If percentage >98.5%, it is exactly matching.
If percentage >95%, it is roughly matching.
When two images generate a new image, there might be some errors. In most cases, my agent uses rough matches.

My agent selects only the exact correct answer. It does not rate answers on a more continuous scale because the agent does not score the transformation.

What mistakes does your agent make? Why does it make these mistakes? Could these mistakes be resolved within your agent's current approaches or are they fundamental problems with the way your agent approaches these problems?

My agent can quickly and correctly solve 12 basic problems D and E. These problems generally have very common transformations. For these problems, visual methods are good enough. But when my agent tries to solve 12 challenge problems D and E, only 7 challenge problems D and only 3 challenge problems E are correct. These challenge problems are much more complex. My agent only implements some simple visual identification. For example, my agent cannot figure out the detailed shape and orientation of objects in images. Let us see the following problem.



My agent cannot find a right production rule to get an answer (option6). My agent cannot figure out polygons, which is quadrilateral, which is pentagon, which is hexagon, and so on.

My agent's current approach cannot resolve these mistakes unless my agent can improve visual capture.

In reality, my agent will meet many more complex problems. When my agent solves these problems, my agent must strengthen images processing capabilities and provide better visual methods.

What improvements could you make to your agent given unlimited time and resource? How would you implement those improvements? Would those improvements improvement your agent's accuracy, efficiency, generality or something else?

Given unlimited time and resources, I make much improvement to my agent. My agent has limited visual methods. First my agent need further improve visual methods. My agent is designed to handle problems that have basic transformations. My agent plans to capture visual information to generate new production rules. My agent needs constraints pattern to handle complex visual images. For often-encountered problems, the agent has more production rules and can provide the right answer in a short time.

1.Orientations of one or more objects are same in each row or each column, but changes down a column or right a row.
2.A regular increase or decrease between adjacent cells in shape.
3.Three values of a category such as shape and color are always present in one row or one column.

Second, my agent tries to use other visual methods to improve efficiency and accuracy. My agent wants to apply fractal method. The fractal method seeks to find a re-representation of images within a problem as a set of similitude transforms at a **significantly finer partitioning** of the images. It assumes elements with a row or a column are related by similitude transformations. A measure of similarity S between the candidate transform C and the target transform T is calculate using the following formula.

$$\textbf{Similarity (T,C)=}\boldsymbol{f}(\boldsymbol{T}\cap\boldsymbol{C})/(\boldsymbol{f}(\boldsymbol{T}\cap \mathrm{C})+\boldsymbol{\alpha f}(\boldsymbol{T}-\boldsymbol{C})+\boldsymbol{\beta f}(\boldsymbol{C}-\boldsymbol{T}))$$

It discovers which similitude transformation best fits any of the complete rows or columns. The answer with the highest calculated similarity is correct.

Third, my agent wants to build a database to store past cases as a long-term memory. The agent can apply generate & test on problems and get an answer. It might store that problem and solution as a case in the database. The on future problems, my agent might find a close match between new problem and old case in the database. My agent can apply that case’s reasoning instead of generate & test method.

Definitely, I think if these improvements are implemented, my agent’s accuracy, efficiency and generality will be improved greatly.

How well does your agent perform across multiple metrics? Accuracy is important, but what about efficiency? What about generality? Are there other metrics or scenarios under which you think your agent’s performance would improve or suffer?

My agent did very well across multiple metrics. For 12 basic problems D and E, all answers are correct. So my agent has very high accuracy. My agent uses some very common production rules. My agent has limited visual processing capabilities and only considers very limited production rules. For 12 challenge problems D and E, most answers are not correct. Seen in this light, my agent has limited generality.

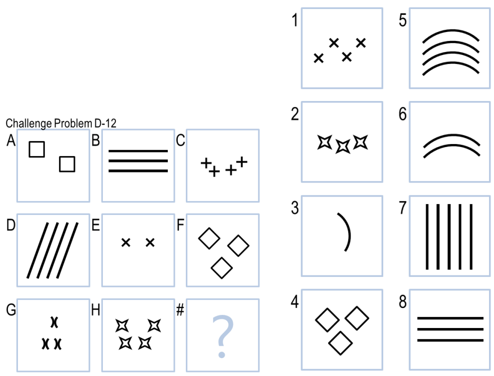
In limited time, my agent is not concise. It carries out some repetitive work that decreases efficiency. And because of lack of database to save plenty of past cases, my agent’s efficiency is not outstanding.

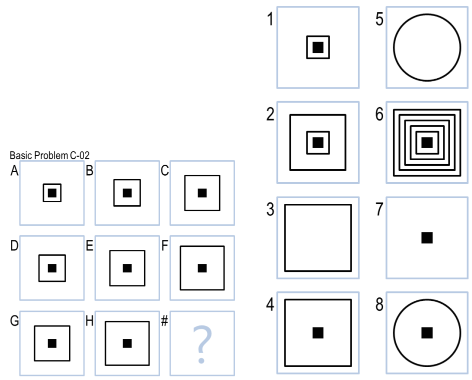
In order to improve my agent’s performance, other production rules mentioned above also should be considered. Other visual methods need to be used, such as a fractal method decomposing an image into a set of smaller images to probably greatly improve accuracy and generality.

Which reasoning method did you choose? Are you relying on verbal representation or visual? If you’re using visual input, is your agent processing it into verbal representations for subsequent reasoning, or is it reasoning over images themselves?

My agent builds a production system in which knowledge is represented as “rules”. This production system includes episodic knowledge, semantic knowledge and procedural knowledge. Procedural knowledge containing production rules is very important because it uses rule-based reasoning to solve problems. And my agent also tries to learn a rule that breaks an impasse by invoking episodic knowledge.

Identification of false instance is important. So my agent needs use diagnosis to investigate when it made an incorrect answer.
An example is shown below.





Rule: The number of objects in each column is equal.
Based on this rule, answer is image 2 that is correct. But if this rule is applied to Basic Problem C-02, the answer could be false. So my agent further constrains this rule further to avoid false answers.

In project 3 my agent only relies on visual representations. And my agent cannot process visual representations into verbal representations.

Finally, what does the design and performance of your agent tell us human cognition? Does your agent solve these problems likes a humans does? How is similar and how is it different? Has your agent's performance given you any insights into the way people solve these problems?

reasoning + learning + memory = deliberation

Production system is a model of human cognition. My agent maps the working memory (current problem) to production rules and decides on an action. Production system (especially its chunking example) is a good example of "learning". At the same time, my agent uses diagnosis to modify production rules. Diagnosis is also a common cognitive task. Whenever a discrepancy between the expected behavior and observed behavior, diagnosis begins.

We cannot have a complete and correct knowledge system. We only have limited computational resources. For different scenarios, we do not always have resource to ensure the reasoning is correct. So we also need come up with potential solutions to a problem, and test the solutions out.

My agent uses observations from visual representations and forms basic production rules. Then the agent maps these rules and decides on an action. The answer that matches the closest rule is picked up. A human does a same procedure when he solves a problem. But a human with a long memory, good intuition and good creative ability has deeper knowledge and better analysis ability. My agent has not.

My agent gives me many insights into the way people solve these problems. When I try to solve any problem, I analyze its working memory, production rules/episodic memory to decide actions. Maybe I first start with no rules, but induce some rules that govern transformation among some images (A->B->C), and transfer these rules to other images (G->H->answer) to get an answer. When I make an incorrect answer, I need figure out which data leads to an incorrect answer, which rules lead to an incorrect answer, how do we modify the rule we pick up to avoid making the same answer, and so on.

In short, it makes me understand deeply how we think and reason using knowledge when we try to solve complex problems.