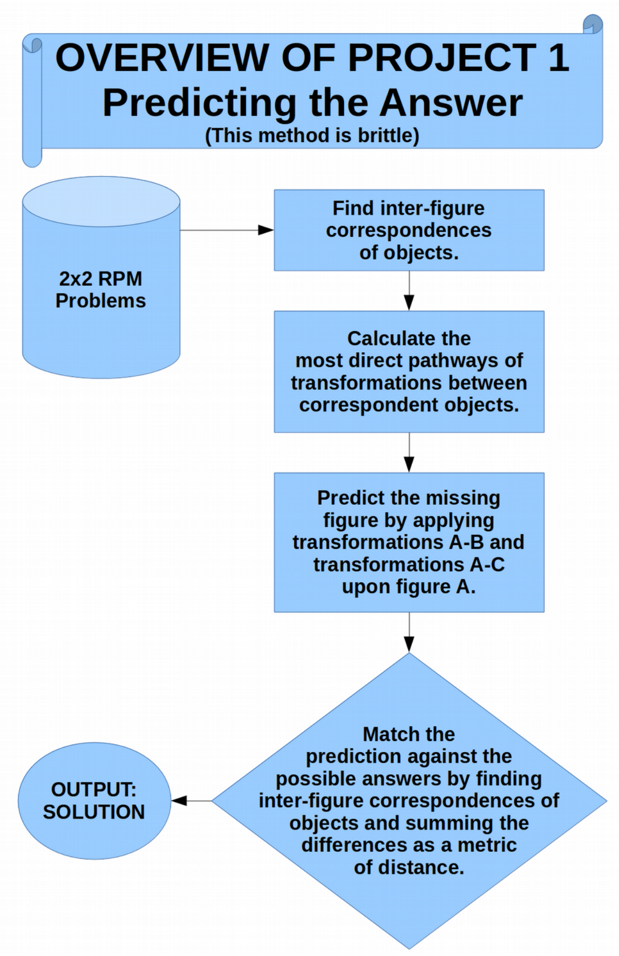
Knowledge-Based Artificial Intelligence – Cognitive Systems (Spring 2017)
Juan Carlos Kuri Pinto, GTid: 903271649
jckp3@gatech.edu, jckuri@gatech.edu, jckuri@gmail.com

Project Reflection 3: Solving 3x3 RPM using Visual Representations

Abstract: This essay abstractly explains the implementation of Project 3 which is about solving 3x3 RPM using visual representations. This implementation is based on the **Fractal Method** (McGreggor, K., Kunda, M., and Goel, A., 2014) and on the tools of the course Knowledge-Based Artificial Intelligence – Cognitive Systems. (Goel, A. and Joyner, D., 2015) And it also has connections to some basic concepts of neuroscience.

Project 1 & 2 Overview

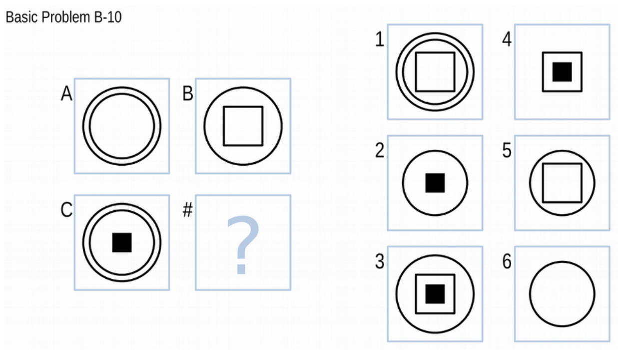
Project 1 only uses **verbal** representations to solve problems. Verbal reasoning is significantly faster, easier to program, and easier to test than visual reasoning. Here is the overview of Project 1:



This implementation of Project 1 produced the following results:

Problem Set	Correct	Incorrect	Skipped
Basic Problems **B** (Verbal/visual and seen)	**12**	**0**	**0**
Challenge Problems **B** (Visual and seen)	0	0	12
Test Problems **B** (Verbal/visual and **unseen**)	**10**	**0**	**2**
Ravens Problems **B** (Visual and **unseen**)	0	0	12

Here is an example to see how Project 1 reasons:



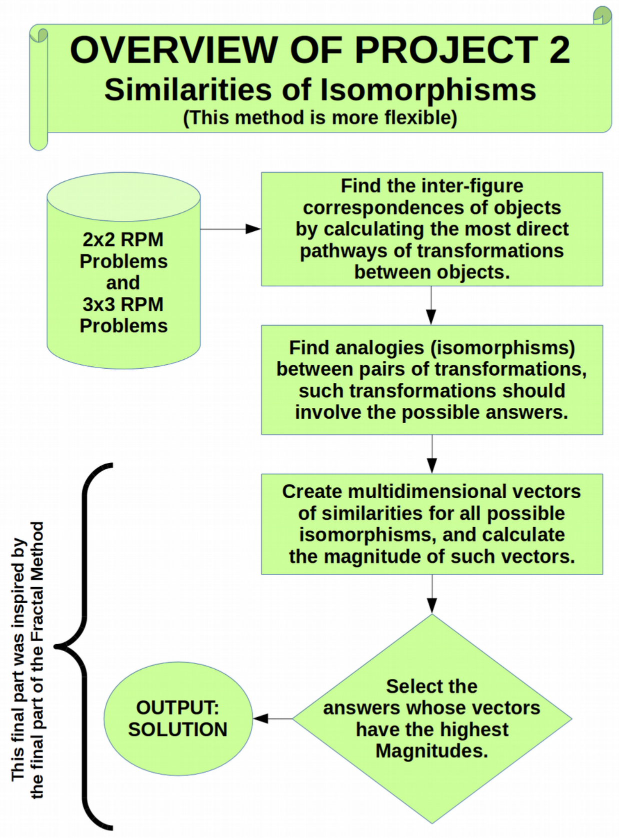
When it comes to horizontal transformations, aka transformations between columns, between A and B, the outer circle remains invariant. Whereas the inner circle is transformed into a smaller square. In like manner, between C and #, the outer circle and the black square should remain invariant. Whereas the inner circle should be transformed into a smaller square, suggesting the correct answer is #3, which is the most similar match.

When it comes to vertical transformations, aka transformations between rows, between A and C, both circles remain invariant. But a black square is added. In like manner, between B and #, the outer circle and the smaller square should remain invariant. But a black square should be added, suggesting the correct answer is #3, which is the most similar match.

Project 1 tries to infer the transformations in a very precise way because the prediction based on such inferred transformations needs to be very precise. But such method is prone to errors because the inferred transformations could be ambiguous or imprecise. Thus, the prediction could also be ambiguous or imprecise, producing errors. This approach is brittle.

Project 2 tries to infer the transformations in an abstract way because no predictions are required, but just counting the differences and similarities between abstract transformations. This approach is more robust, more general, and more flexible like analogies, which are the building blocks of animal intelligence.

Project 2 only uses **verbal** representations to solve problems. Here is the overview of Project 2:



The metric to measure similarity between transformations is the Tversky index:

$$S(X,Y)=\frac{f(X\cap Y)}{f(X\cap Y)+\alpha\cdot f(X-Y)+\beta\cdot f(Y-X)}$$

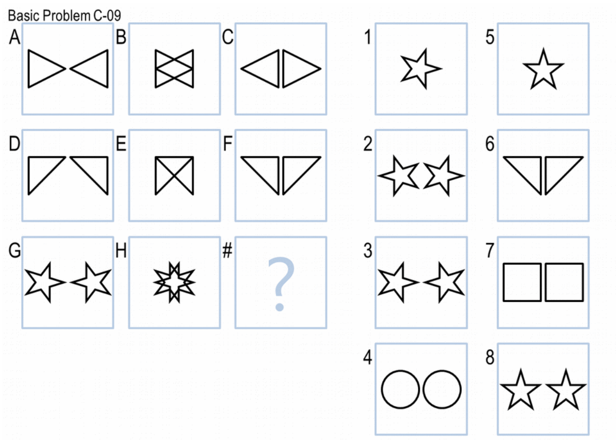
Where f(X) is the number of features found in set X. X and Y can be sets of transformations like A:B :: C:D. So, X can be A:B and Y can be C:D. Tversky index needs to calculate 3 sets of transformations: The intersection of common transformations in A:B and C:D. The transformations found in A:B but not in C:D. And the transformations found in C:D but not in A:B. This is an asymmetric metric when alpha=1 and beta=0. This asymmetry requires to also consider an additional permutation in the order of the isomorphisms like this: C:D :: A:B.

This implementation of Project 2 produced the following results:

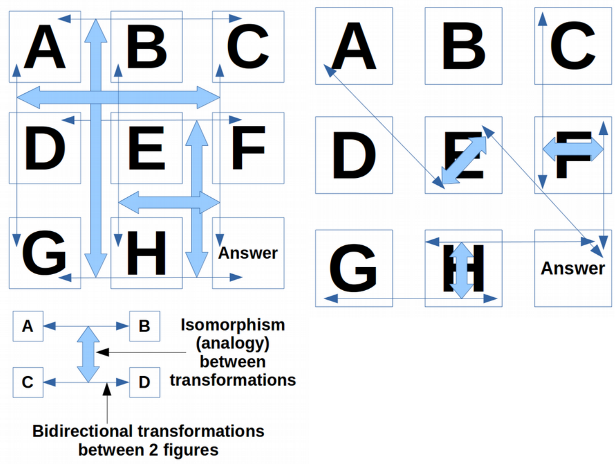
Problem Set	Correct	Incorrect	Skipped
Basic Problems **B** (Verbal/visual and seen)	**12**	**0**	**0**
Challenge Problems **B** (Visual and seen)	0	0	12
Test Problems **B** (Verbal/visual and **unseen**)	**11**	**1**	**0**
Ravens Problems **B** (Visual and **unseen**)	0	0	12

Problem Set	Correct	Incorrect	Skipped
Basic Problems **C** (Verbal/visual and seen)	**10**	**2**	**0**
Challenge Problems **C** (Visual and seen)	0	0	12
Test Problems **C** (Verbal/visual and **unseen**)	**11**	**1**	**0**
Ravens Problems **C** (Visual and **unseen**)	0	0	12

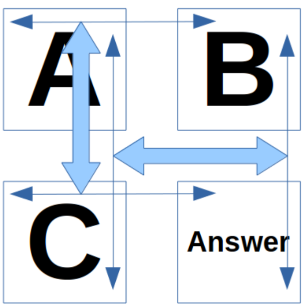
Here is an example to see how Project 2 reasons:



The agent finds the following analogies (isomorphisms):



For the case of 2x2 matrices, the agent finds the analogies:



Getting back to the example, this process finds the inter-figure correspondences. In this case, the agent knows that the 2 figures, which cross each other when transforming in the horizontal axis, are correspondent. The agent also knows shapeshifting, when transforming in the vertical axis, does not affect the correspondences of these 2 figures.

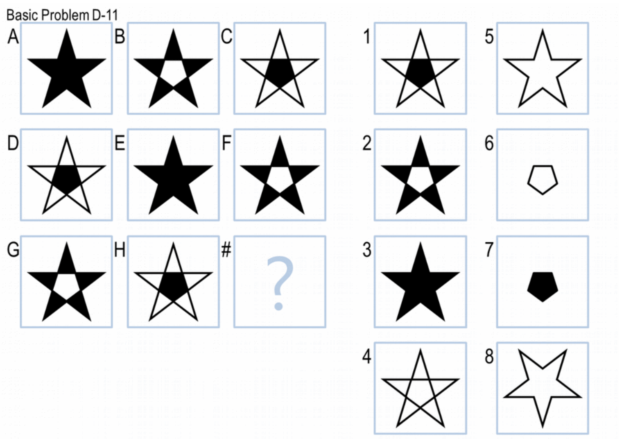
Finally, answer #2 produces the most similar isomorphisms between transformations because these 2 figures cross each other when transforming in the horizontal axis and shift their shapes when transforming in the vertical axis.

Giving the level of abstractions used in this project, it is suggested to study RPM through the lenses of category theory, a branch of abstract algebra that deals with isomorphisms. (Mac Lane, S., 1998)

This method to solve RPM is very similar to the definition of analogies in the book "Symmetry Rules: How Science and Nature Are Founded on Symmetry". In this book, analogies are expressed with the mathematics of symmetry, which are abstract algebra. Analogy is symmetry because when entities change, their relationships remain invariant. Symmetry is invariance under transformations. (Rosen, J., 2008)

Changes in Problem Set and Agent's Reasoning

The problems found in Problem Sets D and E are harder because they have more complex transformations between figures and more complex relationships between objects. Such problems also have periodic transformations between figures that rotate at the matrix level like this:



This kind of periodic transformations between figures that rotate at the matrix level are dealt with the MutualFractal() operator that allows to capture binary, ternary, or n-ary relationships between figures by representing them as permuted features (both specific and agnostic) stored in big hashtables. Thus, the matrix-level position of such inter-figure transformations does not matter because the hashtables retrieve them correctly regardless of their matrix-level position due to the set-theoretic properties of hashtables. Brains also recognize patterns regardless of their translations, aka translational invariance.

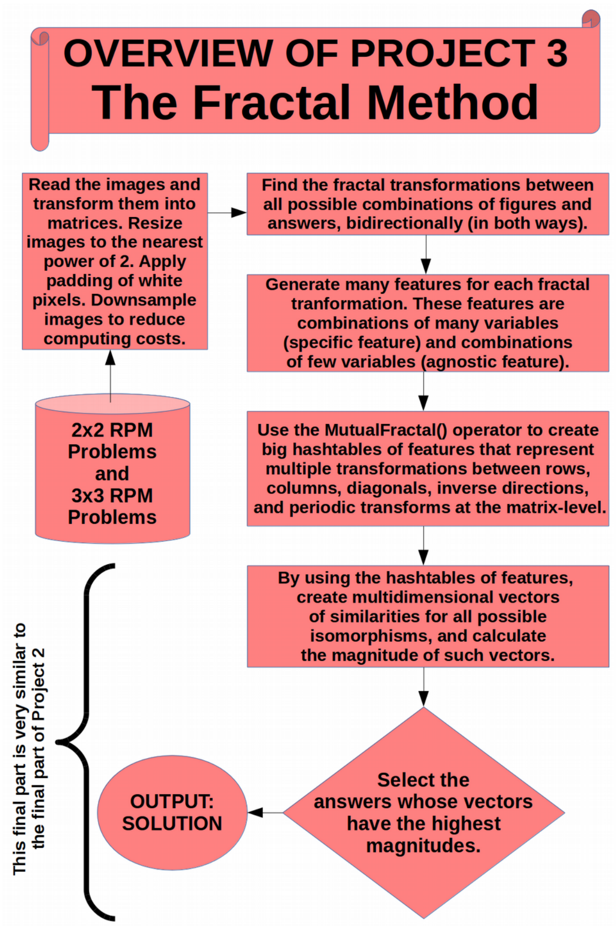
Moreover, Problem Sets D and E only have visual representations, obliging students to program agents capable of reasoning visually. Project 1 and Project 2 only used verbal representations. So, Project 3 required a redesign to reason with visual representations. Since previous problems also had visual representations, such redesign of Project 3 should not affect the resolution of previous problems.

Project 3 only uses **visual** representations to solve problems and is based on the Fractal Method. (McGreggor, K., Kunda, M., and Goel, A., 2014) McGreggor's implementation of the Fractal Method produced impressive results that match the average human performance in all problem sets.

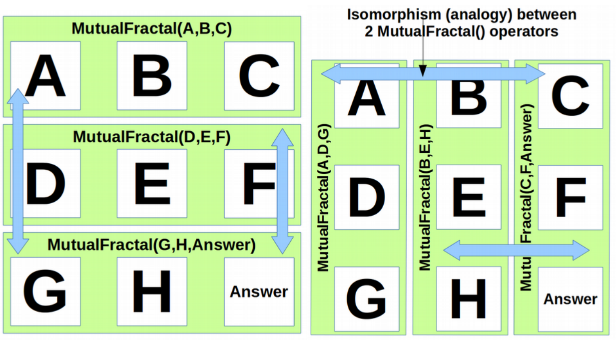
Agent Capabilities

The agent of Project 3 is based on the Fractal Method which reasons visually. Since all RPM problems have visual representations, this new agent can deal with the previous problems and the new problems.

The Fractal Method is too complex to be completely explained in this essay. In fact, it was the Ph.D. dissertation of Prof. McGreggor. (McGreggor, K., 2013) That's why only a brief summary of the Fractal Method will be provided. Here is the brief overview of Project 3:



In addition to the Tversky similarity and the isomorphisms used by the agent of Project 2, the agent of Project 3 also uses the following isomorphisms:



For the case of 2x2 matrices, the agent uses the following isomorphisms:

- MutualFractal(A,B) :: MutualFractal(C,Answer)
- MutualFractal(A,C) :: MutualFractal(B,Answer)

The MutualFractal() operator allows to deal with the new challenge imposed by Problem Sets D and E: Periodic transformations between figures that rotate at the matrix level.

The hardest-to-program and most computationally expensive module of this project was finding the fractal transformations between figures. That's why the source code of the following demo was released; so that future students of the KBAI class can download it from this link. (Kuri, J., 2017) This demo programmed in Python was based on the fractal image compressor in this link. (Kennberg, A., 2013)

In this demo, the fractal transformations of the batman logo are found. In the first row, the source image, the source image downsampled to half the size, and the target image are shown from left to right. In the second row, an arbitrary image (the squared spiral) is gradually and iteratively transformed into the batman logo, which is the fractal attractor of the transformations found. Any arbitrary image will gradually converge to the fractal attractor. The best results in the fractal reconstruction are obtained when the block size is 2. When the block size is 4, the fractal reconstruction is imperfect, making the agent kind of blind.

Fractal image compression work well when the source image and the target image are the **same**. Whereas the Fractal Method for solving Raven's progressive matrices use different images. In this case, the goal is to compute transformations and analogies between transformations. The goal is not to compress and to reconstruct fractal images. Here is the disaster that occurs when the source image and the target image are **NOT** the same:

These programming experiments are important to understand the limitations of the agent. After all, fractal image compression is not magical. Try to remember what happens when the block size is 4 because the agent used a block size of 4.

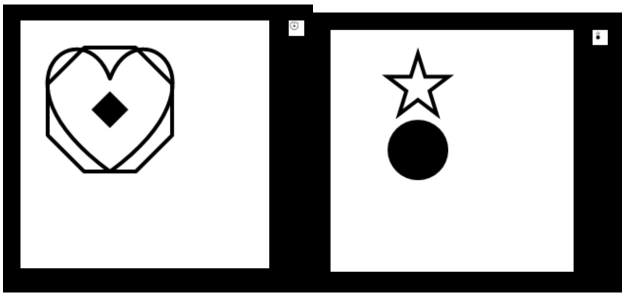
Agent Limitations

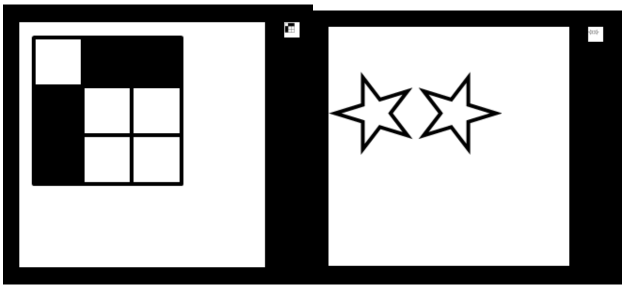
The main limitation of the Fractal Method is the fact that fractal image compression is too computationally expensive. This fact prevented fractal image compression from becoming the Internet standard for sharing images. Other file formats like JPG and PNG won this tough competition. The same applies to the case of videos.

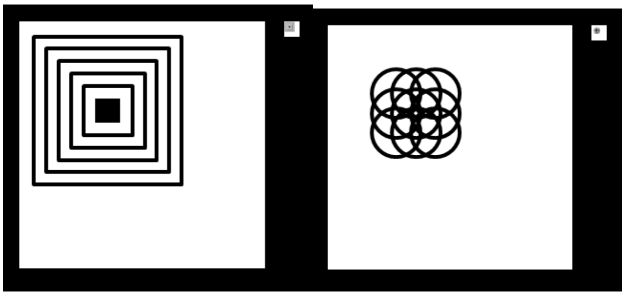
The agent used a size of 16 pixels and a block size of 4 pixels when it was submitted. The submission process lasted 1652.53 seconds (27.5 minutes). By doubling the size or by halving block size, the time of the algorithm multiplies by 4, or even more. In other words, using a size of 32 pixels and a block size of 4 pixels will take 2 hours approximately which is 30 minutes more than the time limit: 90 minutes. And using a size of 16 pixels and a block size of 2 pixels will take 2 hours approximately. Both configurations were tested without success:

```
python submit.py --provider gt --assignment P3
GT Login required.
Username :jckp3
Password :
Save the jwt?[y,N]n
{
    "error": "{\"msg\": \"The command $(sudo -H -u vmuser_xnvkpaos bash -c \\\"cd /home/vmuser_xnvkpaos; ulimit -f 160000 ; ulimit -c 10000 ; python run.py P3 1> run_stdout.txt 2> run_stderr.txt\\\") exceeded the timeout of 5400 seconds.\"}"
}
```

All figures were preprocessed. First, images were resized to the nearest power of 2, which is 256 pixels. Padding of white pixels was applied. Then, images were downsampled 4 times. Each downsampling halves the size of images. So, the resulting images have a size of 16. Here are some examples of the original images (256x256) and the tiny downsampled images (16x16) at the right of the original images.







The tiny downsampled images are almost invisible. This is a big disadvantage for the agent.

The agent used a block size of 4 which cannot reconstruct images in a perfect way. This is another disadvantage. Remember the experiment with a block size of 4:

The agent could not see clearly for these 2 reasons. *Just hoping that instructors will be kind when grading agent's poor performance.*

In the case of problems with verbal representations, Project 2 performs better than Project 3. But Project 3 can face the visual problems that Project 1 and Project 2 cowardly skipped.

Another limitation is the fact the Fractal Method lacks visual common sense, that is, the accumulated experiences acquired in a lifetime (ontogeny) and throughout the evolution of the species (phylogeny). Some RPM problems requires common sense reasoning.

Agent Performance

This implementation of Project 3 produced the following results in submission #2 and submission #3:

Problem Set	Submission #2		Submission #3	
	Correct	Incorrect	Correct	Incorrect
Basic Problems **B** (Verbal/visual and seen)	**5**	7	**6**	6
Challenge Problems **B** (Visual and seen)	2	10	2	10
Test Problems **B** (Verbal/visual and **unseen**)	**6**	6	**8**	4
Ravens Problems **B** (Visual and **unseen**)	**6**	6	**5**	7

Problem Set	Submission #2		Submission #3	
	Correct	Incorrect	Correct	Incorrect
Basic Problems **C** (Verbal/visual and seen)	3	9	1	11
Challenge Problems **C** (Visual and seen)	0	12	1	11
Test Problems **C** (Verbal/visual and **unseen**)	1	11	**4**	8
Ravens Problems **C** (Visual and **unseen**)	1	11	1	11

Problem Set	Submission #2		Submission #3	
	Correct	Incorrect	Correct	Incorrect
Basic Problems **D** (Visual and seen)	**4**	8	2	10
Challenge Problems **D** (Visual and seen)	2	10	1	11
Test Problems **D** (Visual and **unseen**)	3	9	3	9
Ravens Problems **D** (Visual and **unseen**)	2	10	3	9

Problem Set	Submission #2		Submission #3	
	Correct	Incorrect	Correct	Incorrect
Basic Problems **E** (Visual and seen)	1	11	0	12
Challenge Problems **E** (Visual and seen)	1	11	0	12
Test Problems **E** (visual and **unseen**)	0	12	0	12
Ravens Problems **E** (Visual and **unseen**)	2	10	**4**	8

In spite of the problems previously explained, the agent obtained a decent performance.

In 2x2 RPM problems, the agent obtained some scores above 5/12. The agent even obtained 8/12 in the **unseen** Test Problems B. Thus, the agent generalizes well.

In 3x3 RPM problems, the agent obtained some scores of 4/12, which are 2.5 points above the random score of 1.5. (12 problems * P(random correct answer) = 12*1/8 = 1.5)

With a size of 64 pixels and a block size of 2, the agent would surely obtained much better results than with the almost blind configuration used in the submissions. Each submission with the blind configuration took 28 minutes approximately. Using the suggested configuration would require many hours, probably more than 1 day.

A way to overcome this issue is to use the Fourier transform. For example, the KBAI classmate Bogdan Vatulya has successfully applied the Discrete Wavelet Transform to accelerate his submission below the time limit. Prof. Michael Barnsley created a new mathematical tool called Fractal Fourier Analysis which is very fast. (Barnsley, M., 2015) And Prof. Jürgen Schmidhuber used the Fast Fourier Transform to accelerate the learning phase of neural networks by many orders of magnitude. (Schmidhuber, J., 2013)

Agent's Relation to KBAI

Due to the visual nature of the Fractal Method, it is harder to see a clear connection between this agent and KBAI tools, which are mostly symbolic. But it is possible to spot some connections.

Frames and **Semantic Networks** are deeply intertwined. Frames have attributes that can point to other frames, generating complex networks of objects, which are semantic networks. Each Python object has many fields just like frames have many attributes. In fact, object orientation, frames, and semantic networks were born at the same time and share the same origins. The Fractal Method uses complex networks of Python objects. Thus, it is based on frames and semantic networks.

Generate & Test is exploited at these stages of processing:

- when searching for the most likely transformations between figures;
- and when searching for the most likely isomorphisms between pairs of transformations.

Problem Reduction is applied when the project is decomposed into many submodules.

Means-Ends Analysis is not applicable to all problems because solving some problems involves trade-offs between the short-term and the long-term. Some decisions are both “bad for the short-term” and “perfect for the long-term”. Means-Ends Analysis is short-sighted and greedy because it only sees rewards for the short-term. Means-Ends Analysis is the greedy version of **Planning**.

Fortunately, transformations between figures are done in parallel. Each visual pattern is only transformed once, not many times. This fact allows to apply Means-Ends Analysis to RPM. Because the problems presented do not show interdependences between different visual patterns. Each visual pattern can be transformed independently. This aspect was more evident in Project 1 and Project 2. But it is also applicable to the Fractal Method.

Logic is applied in each control structure of Python: Conditionals, loops, and data structures. Logic is also applied when the agent performs transformational operations to reason about the problem.

Understanding is used because a set of constraints (visual patterns) help to disambiguate and to find the most likely transformations between figures and the most likely isomorphisms between pairs of transformations. Such transformations and isomorphisms are the agent’s understanding of the problem per se.

Analogical Reasoning is done at 2 complexity levels of abstraction:

- when searching for transformations between figures;
- and when searching for isomorphisms between pairs of transformations.

These are high-level invariances. And analogies are invariant relationships that hold when the entities change. Analogies are the building blocks of cognition. And the Fractal Method is a tentative approach to make artificial analogies, which makes machines have more flexibility to reason and to create.

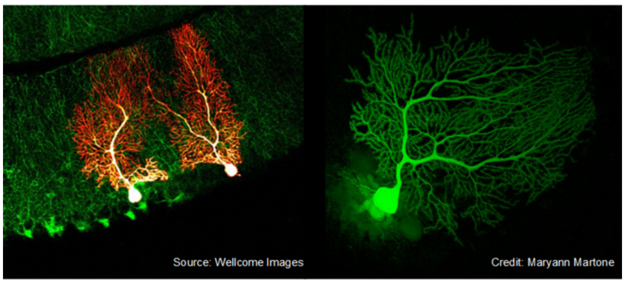
Computational Creativity is the byproduct of **Analogical Reasoning**. When machines have the ability of making analogies, they are able to find new pathways of patterns that can be creative solutions to problems. Creativity is not only about to create never-seen-before patterns. Creativity is rather a basic cognitive function without which intelligent behavior is impossible to achieve.

Agent's Relation to Human Cognition

What are the relationships between fractals and brains? Here are some fractal aspects of brains.

Dendritic and axonal trees of neurons are fractals that maximize their surface of information connectivity. For example: Pure fractals drawn in 2D have a limited area and an infinite perimeter. Their fractal dimension is between 2 and 1. Pure fractals represented in 3D have a limited volume and an infinite surface. Their fractal dimension is between 3 and 2. (Lesmoir-Gordon, N., Rood, W., and Edney, R., 2014)

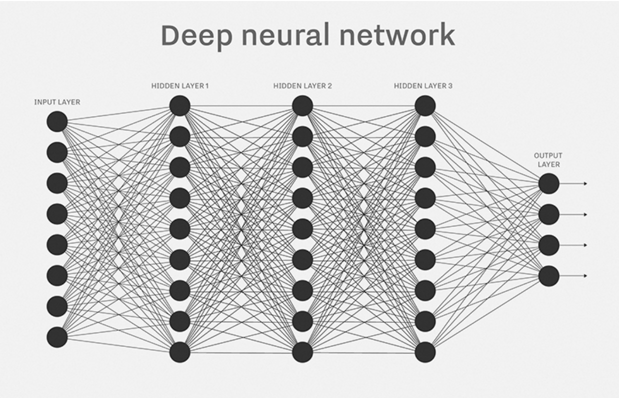
Purkinje cells of the cerebellum. (Deleniv, S., 2016)



Brains are not the only organs of the body that exploit fractal structures to maximize the area of contact. Lungs, the circulatory system, the peripheral nervous system, and the digestive system also exploit the fractal properties of rough structures to maximize their efficiencies. (Lesmoir-Gordon, N., Rood, W., and Edney, R., 2014)

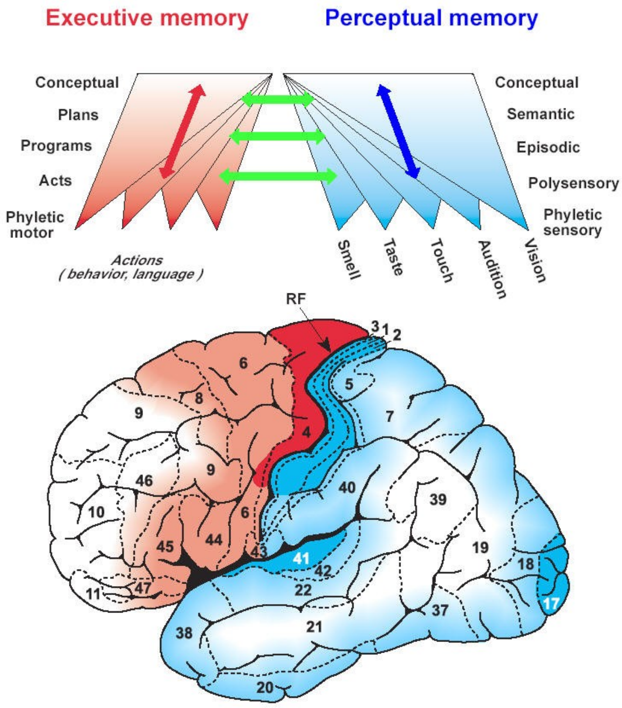
Neural networks are all interconnected (maximal information connectivity) so that all possible correlations of patterns could be captured. Correlation does not necessarily imply causation. But correlations are the precursors of causations. And brains try to learn the causations of reality.

Deep neural network. (Rouse, M., 2016)



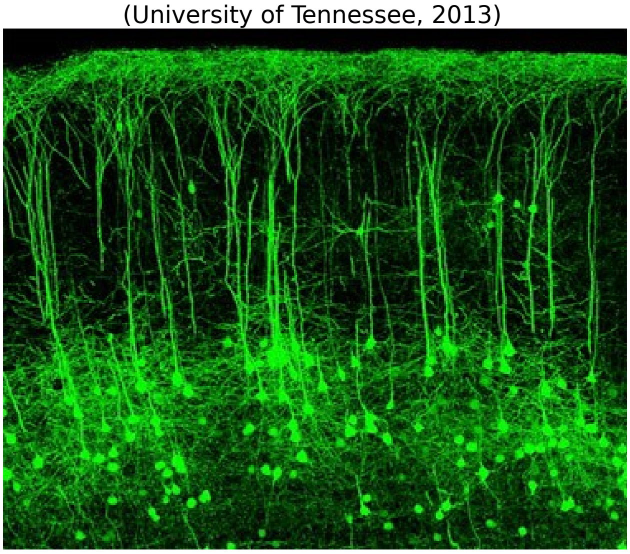
Cortical convolutions are fractals that maximize cortical area in a reduced volume. More cortical area means more cortical columns, aka neural processors. (Kurzweil, R., 2005)

Heterarchical connectivity of the human brain. (Fuster, J., 2007)



Moreover, cognits (cortical columns) have the same connectivity pattern regardless of the modality and the level of complexity they represent. Therefore, they are self-similar at many levels of complexity like fractals. (Hawkins, J. and Blakeslee, S., 2004)

Cortical columns of a transparent brainbow with green luminescent firings in GMO mouse (University of Tennessee, 2013)



Finally, analogies are the basic building blocks of cognition. McGreggor's Fractal Method makes artificial analogies. However, it is still speculative to suggest brains make analogies with a similar mechanism. More investigation is needed.

The actual role of cortical columns is being investigated in an intense way. But, due to their highly convoluted and deeply intertwined nature, cortical columns probably try to find analogies, transformations, and correspondences of the world. Correspondences between sensations and percepts. Between raw sensory data and the laws of physics. The brain tries to find the invariants of the world. (Hofstadter, D. and Sander, E., 2013)

In like manner, this KBAI agent tries to find the analogies, transformations, and correspondences of Raven's Progressive Matrices (RPM). Why are people considered intelligent when they solve RPM? Because they can understand the high-level patterns in these visuospatial problems. They can understand the rules of such transformations.

In spite of the simplicity of this KBAI agent, its structure is quite convoluted and interrelated like brains are. It has many nested for-loops that try to find all the possible combinations of patterns in order to find the most appropriate answers for the problems.

If a map of the relationships between function calls, variable assignments, operations, loops and other control structures of this agent were drawn, the result would be a very convoluted and interrelated network like the one formed by cortical columns and the connectome. It would be very difficult to draw such a mess in 2D.

References

McGreggor, K., Kunda, M., and Goel, A. (2014). Fractals and Ravens. Retrieved from: http://www.sciencedirect.com/science/article/pii/S0004370214000587

Goel, A. and Joyner, D. (2015). CS 7637: Knowledge-Based Artificial Intelligence: Cognitive Systems. Georgia Institute of Technology. Retrieved from:
https://www.omscs.gatech.edu/cs-7637-knowledge-based-artificial-intelligence-cognitive-systems

Mac Lane, S. (1998). Categories for the Working Mathematician. Second Edition. Springer.

Rosen, J. (2008). Symmetry Rules: How Science and Nature Are Founded on Symmetry. Springer.

McGreggor, K. (2013). Fractal Reasoning. Retrieved from:
https://smartech.gatech.edu/bitstream/handle/1853/50337/MCGREGGOR-DISSERTATION-2013.pdf

Kuri, J., (2017). FractalCompressor.py. Retrieved from:
https://pastebin.com/69SLPFkJ

Kennberg, A. (2013). kennberg/fractal-compression. Retrieved from:
https://github.com/kennberg/fractal-compression

Barnsley, M. (2015). Fractal Fourier Analysis. Retrieved from:
http://superfractals.com/wpfiles/fractal-fourier/

Schmidhuber, J. (2013). Compressed Network Search. Retrieved from:
http://people.idsia.ch/~juergen/compressednetworksearch.html

Lesmoir-Gordon, N., Rood, W., and Edney, R. (2014). Introducing Fractals: A Graphic Guide. Icon Books Ltd.

Deleniv, S. (2016). Capturing a Beautiful Neuron. Retrieved from:
https://theneurosphere.com/2016/01/

Rouse, M. (2016). Definition of Neural Network. Retrieved from:
http://searchnetworking.techtarget.com/definition/neural-network

Kurzweil, R. (2005). The Singularity Is Near: When Humans Transcend Biology. Penguin Books.

Fuster, J. (2007). Scholarpedia - Cortical memory. Retrieved from:
http://www.scholarpedia.org/article/Cortical_memory

Hawkins, J. and Blakeslee, S. (2004). On Intelligence. Times Books.

University of Tennessee. (2013) Neuroscience Institute - Imaging Center. Retrieved from:
https://www.uthsc.edu/neuroscience/imaging-center/

Hofstadter, D. and Sander, E. (2013). Surfaces and Essences: Analogy as the Fuel and Fire of Thinking. Basic Books.