

# A Structure-Mapping Model of Raven's Progressive Matrices

**Andrew Lovett (andrew-lovett@northwestern.edu)**

**Kenneth Forbus (forbus@northwestern.edu)**

**Jeffrey Usher (usher@cs.northwestern.edu)**

Qualitative Reasoning Group, Northwestern University, 2133 Sheridan Road  
Evanston, IL 60208 USA

## Abstract

We present a computational model for solving Raven's Progressive Matrices. This model combines qualitative spatial representations with analogical comparison via structure-mapping. All representations are automatically computed by the model. We show that it achieves a level of performance on the Standard Progressive Matrices that is above that of most adults, and that the problems it fails on are also the hardest for people.

**Keywords:** Analogy, Spatial Cognition, Problem-Solving

## Introduction

There is increasing evidence that visual comparison may rely on the same structural alignment processes used to perform conceptual analogies (Markman & Gentner, 1996; Lovett et al., 2009a; Lovett et al., 2009b). An excellent task for exploring this is the Raven's Progressive Matrices (RPM) (Raven, Raven, & Court, 2000). In RPM problems (Figure 1), a test-taker is presented with a matrix of images in which the bottom right image is missing, and asked to pick the answer that best completes the matrix. Though RPM is a visual task, performance on it correlates highly with other assessment tasks, many of them non-visual (e.g., Snow & Lohman, 1989; see Raven, Raven, & Court, 2000, for a review). Thus, RPM appears to tap into important, basic cognitive abilities beyond spatial reasoning, such as the ability to perform analogies.

This paper presents a computational model that uses analogy to perform the RPM task, building on existing cognitive models of visual representation and analogical comparison. Our claims are:

- 1) Tasks such as RPM rely heavily on qualitative, structural representations of space (e.g., Biederman, 1987; Forbus, Nielsen, & Faltings, 1991). These representations describe relations between objects in a visual scene, such as their relative location. Importantly, these representations are hierarchical (Palmer, 1977); they can also describe larger-scale relations between groups of objects or smaller-scale relations between parts of an object.

- 2) Spatial representations are compared via structure-mapping (Gentner, 1983), a process of structural alignment first proposed to explain how people perform analogies. Structure-mapping is used here to compute the similarity of two images, to identify corresponding objects in the images, and to generate abstractions based on commonalities and differences.

We previously (Lovett, Forbus, & Usher, 2007) described a model based on these principles that achieved human

adult-level performance on two sections of the Standard Progressive Matrices test. That model was unable to handle the more difficult sections of the test because it only considered differences between pairs of images. This paper describes a more advanced model which performs at an above-average level on the hardest four sections of the test. It remains grounded in the same principles but is able to identify patterns of differences across rows of images. Like before, all inputs are automatically computed from vectorized input.

We first discuss Carpenter, Just, and Shell's (1991) computational model of the RPM. We then describe our model and its results on the Standard Progressive Matrices test. We end with conclusions and future work.

## Background

The best-established model of Raven's Progressive Matrices was developed by Carpenter, Just, and Shell (1991). It was based on both analysis of the test and psychological studies of human performance. The analysis led to the observation that all but two of the problems in the Advanced Progressive Matrices, the hardest version of the test, could be solved via the application of a set of six rules (see Figure 1 for examples). Each rule describes how a set of corresponding objects vary across the three images in a row. The simplest, Constant in a Row, says that the objects stay the same. Quantitative Pairwise Progression (Figure 1A) says that one of the object's attributes or relations gradually changes. The other rules are more complex, requiring the individual to align objects with different shapes (Distribution of Three), or to find objects that only exist in two of the three images (Figure Addition or Subtraction, Distribution of Two).

The psychological studies suggested that most people solved the problems by studying the top row, incrementally generating hypotheses about how the objects varied across that row, and then looking at the middle row to test those hypotheses. This process began by comparing consecutive pairs of images in a row.

Armed with their observations, Carpenter et al. built two computational models to solve the Advanced Progressive Matrices: FAIRAVEN and BETTERAVEN. Both models used hand-coded input representations. They solved a problem by: 1) identifying which of the six rules applied to the first two rows, and 2) computing a mapping between those two rows and the bottom row to determine how to apply the same rules in that row.

	A	B	C
Carpenter Rules	Quantitative Pairwise Progression	Constant in a Row + Distribution of Three	Distribution of Three (applies twice)
Our Classification	Differences	Literal	Advanced Literal
Answer	3	5	2
	D	E	F
Carpenter Rules	Distribution of Three (applies twice)	Figure Addition or Subtraction	Distribution of Two (applies two or three times)
Our Classification	Advanced Literal	Advanced Differences	Advanced Differences
Answer	4	5	7

Figure 1: Several examples of RPM problems. To protect the security of the test, all examples were designed by the authors. Included are the rules required to solve the problems according to Carpenter et al.'s (1991) classifications.

BETTERAVEN differed from FAIRAVEN in that it possessed better goal-management and more advanced strategies for identifying corresponding objects in a row. Whereas FAIRAVEN could perform at the level of the average participant in their subject pool, BETTERAVEN matched the performance of the top participants.

Since BETTERAVEN's development, studies (Vodegel-Matzen, van der Molen, & Dudink, 1994; Embretson, 1998) have suggested that Carpenter et al.'s rule classification is a strong predictor of the difficulty of a matrix problem: problems that involve the more advanced rules, and that involve multiple rules, are more difficult to solve. In this respect, the models have had an important, lasting legacy. Unfortunately, they have two limitations. First, they operate on hand-coded input, hence the problem of generating the spatial representations is not modeled. Carpenter et al. justify this by pointing to the high correlation between RPM and non-spatial tasks, suggesting that perceptual encoding must not play an important role in the task. However, an alternate explanation is that the problem of determining the correct spatial representation for solving a matrix relies on encoding and abstraction abilities shared with other, non-visual modalities. The second drawback is that the six rules

identified by Carpenter et al. were hard-coded into their models. Thus, the models tell us little about how people discover those rules in the first place. That is, how do people progress from comparing pairs of images to understanding how objects vary across a row?

Our model addresses these limitations by using existing models of perceptual encoding and comparison. Spatial representations are automatically generated using the CogSketch (Forbus et al., 2008) sketch understanding system. These representations are compared via the Structure-Mapping Engine (SME) (Falkenhainer, Forbus, & Gentner, 1989) to generate representations of the pattern of variance across a row. We describe each of these systems, beginning with SME as it plays a ubiquitous role in our models of perception and problem-solving.

### Comparison: Structure-Mapping Engine

The Structure-Mapping Engine (SME) (Falkenhainer, Forbus, & Gentner, 1989) is a computational model of comparison based on Gentner's (1983) structure-mapping theory. It operates over structured representations, i.e., symbolic representations consisting of entities, attributes, and relations. Each representation consists of a set of

expressions describing attributes of entities and relations between entities. For example, a representation of the upper-left image in Figure 1B might include an expression stating that the square **contains** the circle.

Given two such representations, a *base* and a *target*, SME aligns their common relational structure to generate a mapping between them. Each mapping consists of: 1) correspondences between elements in the base and target representations; 2) *candidate inferences* based on expressions in one representation that failed to align with anything in the other; 3) a similarity score between the two representations based on the quantity and depth of their aligned structure. For this model, we normalize similarity scores based on the overall size of the base and target.

SME is useful in spatial problem-solving because a mapping between two spatial representations can provide three types of information. First, the similarity score gives the overall similarity of the images. Second, the candidate inferences identify particular differences between the images. Third, the correspondences can be useful in two ways. (a) Correspondences between expressions identify commonalities in the representations, and (b) correspondences between entities identify corresponding objects in the two images, a key piece of information for determining how an object varies across a row of images.

Finally, SME can take as input constraints on its mappings, such as requiring particular correspondences, excluding particular correspondences, or requiring that certain types of entities only map to similar types. While the psychological support for these constraints is not as strong as the overall psychological support for SME, we have found previously (Lovett et al., 2009b) that constraints can be useful for simulating a preference for aligning similar shapes when comparing images.

### Perceptual Encoding: CogSketch

We use CogSketch (Forbus et al., 2008) to generate spatial representations. CogSketch is an open-domain sketch understanding system. Given a sketch consisting of line drawings of a set of objects, CogSketch automatically computes qualitative spatial relations between the objects, generating a spatial representation. This representation can then serve as the input to other reasoning systems.

There are two ways of providing input to CogSketch. A user can either draw out a sketch within CogSketch, or import a set of shapes created in PowerPoint. In either case, it is the user's responsibility to segment an image into objects—CogSketch does not do this automatically.

Essentially, the user is performing part of the job of perceptual organization (Palmer & Rock, 1994), the low-level visual operation that creates a set of entry-level units for processing. We focus on modeling the ways one must

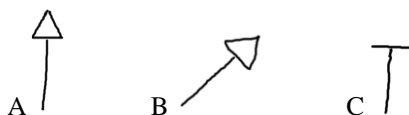


Figure 2. A,B: Two arrow shapes. C: Part of an arrow.

reorganize these units—via grouping and segmentation—during the problem-solving processes.

Sketches can be further segmented by using a *sketch lattice*, a grid which indicates which objects should be grouped together into images. For example, to import the Raven problems in Figure 1 into CogSketch, one would create one sketch lattice for each of the two matrices in a problem, then import the shapes from PowerPoint and place them in the appropriate locations in each lattice. In this way, a user can specify an RPM problem for CogSketch to solve.

### Generating Representations

Given a sketch, CogSketch automatically generates a set of qualitative spatial relations between the objects in it. These relations describe the relative position of the objects and their topology—i.e., whether two objects intersect, or whether one is located inside another. CogSketch can also generate attributes describing features of an object, such as its relative size or its degree of symmetry.

CogSketch is not limited to generating representations at the level of objects. It is generally believed that human representations of space are hierarchical (Palmer, 1977; Palmer & Rock, 1994). While there may be a natural “object” level of representation, we can also parse an object into a set of parts or group several objects into a larger-scale set. Similarly, CogSketch can, on demand, generate representations at two other scales: edges and groups.

To generate an edge-level representation, CogSketch parses the lines that make up an object into edges. It does this by identifying discontinuities in a line's curvature that indicate the presence of corners (see Lovett et al., 2009b for details). CogSketch then generates qualitative spatial relations between the edges in a shape, describing relative orientation, relative length, convexity of corners, etc.

To generate a representation at the level of groups, CogSketch groups objects together based on proximity and similarity. It can then identify qualitative spatial relations between groups, or between groups and individual objects.

### Interactions with SME

We believe structural alignment plays an important role in comparing visual stimuli. CogSketch employs SME to determine how images relate to each other. However, the use of hierarchical representations means that SME can also compare two objects' edge-level representations to determine how the objects relate to each other. Our model uses this capability in two ways, discussed next.

**Finding Shape Transformations** CogSketch can compare two objects' shapes to identify transformations between them, e.g., the rotation between the arrow shapes in Figure 2. It does this via a simple simulation of mental-rotation (Shepard & Metzler, 1971): (1) Two objects' edge-level representations are compared via SME. SME's mapping identifies the corresponding edges in the two objects. (2) Pairs of corresponding edges are quantitatively compared to determine whether there is a consistent transformation

between them. In Figure 2, CogSketch could identify a rotation or a reflection between the arrows shapes.

CogSketch can identify two types of shape transformations: equivalence transformations (henceforth called simply *transformations*) and deformations. Transformations (rotation, reflection, and changes in overall size) leave an object's basic shape unchanged. Deformations (becoming longer/shorter, becoming longer/shorter in a part, adding/losing a part) change the object's shape.

Based on shape comparisons, a given set of objects can be grouped into *equivalent shape classes*—groups of objects that have a valid transformation between them, such as equilateral triangles of all sizes and orientations—and *strict shape classes*—groups of objects that are identical, such as upright, equilateral triangles of a particular size.

**Comparison-Based Segmentation** CogSketch can dynamically segment an object into parts based on comparisons with other objects. For example, to determine the relationship between the images in Figures 2A and 2C, it segments each object into its edges, and uses SME to identify corresponding edges. Grouping only edges in 2A that correspond to edges in 2C enables it to segment 2A into two objects, one of which is identical to 2C. The difference between 2A and 2C is then represented as: *A contains the same object as 2C, but with a second, angular object located above it.*

## Our Model

Our model is based on Carpenter, Shell, and Just's (1991) finding that people generally begin solving a matrix problem by comparing adjacent pairs of images in each row of the problem. Our model begins by comparing the images in a row via SME. Based on the mappings between images, it generates a *pattern of variance*, a representation of how the objects change across the row of images. The model then computes a second-order comparison (Lovett et al., 2009B), using SME to compare the patterns for the top two rows and rate their similarity. If the rows are sufficiently similar, the model builds a generalization representing what is common to them; it then looks for an answer that will allow the bottom row to best match this generalization. If the top two rows are not sufficiently similar, the model makes a change to its problem-solving strategy.

Instead of identifying RPM-specific rules as Carpenter et al. did, we utilize two general classes of strategies (four strategies in all) for how a person might go about building patterns of variance. We believe these strategies should be applicable to a variety of spatial problems.

The two classes of strategies are Differences and Literal. Differences involves representing the differences between adjacent pairs of images in a row. For example, in Figure 1A the object is gradually getting smaller. Literal involves representing what is literally true in each image of the row. In Figure 1B, every row contains a square, a circle, and a diamond. There are also advanced versions of each strategy, described below. We now describe each strategy in detail.

## Differences Strategy

**1) Generate Representations** CogSketch generates a spatial representation for each object in a row. While CogSketch can generate representations at multiple levels, the model begins with the highest-scale, and thus simplest, representation. Objects consisting of a single edge—or objects consisting of multiple edges that don't form a closed shape—are grouped together based on connectedness to form a single object, e.g., in the first image of Figure 1F, the vertical and diagonal edges are grouped to form a single object. Objects consisting of closed shapes are combined based on proximity and similarity to form *groups*, e.g., the sets of three squares in Figure 1F are grouped together. CogSketch then computes spatial relationships between the objects, and between objects and groups. It also computes object attributes, describing their shape, color, texture, etc.

**2) Compute a Basic Pattern of Variance** Consecutive pairs of images in the row are compared via SME to identify the corresponding objects. If there are leftover, unmatched objects in both the first and last images of the row, then these images are also compared. Corresponding objects are then compared to identify transformations between their shapes. Based on these comparisons, the model generates one of the following expressions to describe how an object varies between each pair of images: (a) *Identity*: The object remains the same. (b) *Transformation*: A transformation exists between the shapes. (c) *Deformation*: A deformation exists between the shapes. (d) *Shape Change*: The shapes change entirely. Shape changes are represented as a change between two strict shape classes. Essentially, this is equivalent to a person keeping “square changes to circle” in working memory. (e) *Addition/Removal*: An object is added or removed.

If an object is identical in every image in the row, then this is deemed unimportant, and not explicitly represented<sup>1</sup>. The rest of these expressions are supplemented by any changes in the spatial relations and colors of the images, as identified by SME's candidate inferences, to produce a representation of the pattern of variance across the row.

**3) Comparison-Based Segmentation** For some problems, the appropriate set of objects to consider only becomes clear after images are compared. For example, in Figure 1E, one discovers after comparison that the third object in the row can be segmented into two parts, such that these parts correspond to the previous two objects in the row. Our model attempts comparison-based segmentation for a set of corresponding objects when: (a) The objects can be broken down into edges, i.e., they aren't filled-in shapes. (b) There is at least one total shape change between the objects, suggesting that they currently don't align well. (c) The changed shapes share some similar parts, i.e., edges with

---

<sup>1</sup> Carpenter, Just, and Shell (1991) found that the Constant in a Row rule, in which an object remains identical across a row, did not contribute to the difficulty of problems, suggesting that people simply ignore objects that don't change.

similar lengths and orientations. (d) There are no identity matches between objects.

Comparison-based segmentation is performed by breaking the objects into their edges, comparing their edges in a new pattern of variance, and then grouping the edges back together based on which sets of edges correspond across the images. This approach is key in solving Figure 1E. It also allows the model to determine that the vertical line and “X” shape are separate objects in Figure 1F. A similar approach is used to segment groups into subgroups or individual objects when they misalign.

**4) Compute Final Pattern of Variance** Repeat step 2) after segmentation and regrouping.

### Advanced Differences Strategy

The advanced differences strategy is identical, except that in steps 3-4, SME mapping constraints are used so that objects only map to other objects in the same strict shape class (i.e., identical objects). Additionally, objects consisting of single edges (as when the shapes in Figure 1E are broken down into their edges) can only map to other single-edged objects at the same relative location in the image. This means the model will never find object transformations, but it will often find object additions/removals, making it ideal for solving problems like 1E and 1F, in which each object is only present in two of the images in a row.

### Literal Strategy

The literal strategy represents what is present in each image in a row, rather than what is different between images. It begins by comparing images to identify any features found in all three images (e.g., the inner shapes in Figure 1B). It abstracts these features out, representing only the features in each image that are not constant across the row. If an object has a different shape from other corresponding objects in the row (e.g., the outer shapes in Figure 1B), then the model includes that object’s strict shape class in the representation.

### Advanced Literal Strategy

The advanced literal strategy begins by applying the basic literal strategy. It then removes any references to the images in which the objects are found. Spatial relations between objects are also abstracted out. Thus, each object is represented independently, and allowed to match independently from the other objects in its image (e.g., Figure 1D). Alternatively, if each image contains only a single object, then an object is split up and each of its attributes are represented as a separate entity (Figure 1C).

### Choosing the Best Strategy

Our model evaluates a strategy by computing patterns of variance for the top two rows and using SME to compare them and rate their similarity. If the similarity is above a threshold, the strategy is deemed a success. If not, a different strategy is tried. The strategies are tried in the following order, which approximates simplest to most

complex: Differences, Literal, Advanced Literal, Advanced Differences. If no strategy meets criterion, the model picks whichever Differences strategy receives the highest score—Literal strategies that fail to meet criterion are not considered, since by definition they expect a near-identical match between rows.

### Selecting an Answer

Once a strategy is chosen, the model compares the pattern of variance for the top two rows to construct an analogical generalization (Kuehne et al., 2000), describing what is common to both rows. The model then scores each of the eight possible answers. An answer is scored by inserting that answer into the bottom row, computing a pattern of variance, and then using SME to compare this to the generalization for the top two rows. The highest-scoring answer is selected. In cases of ties, no answer is selected.

### Solving 2x2 Matrices

The easier RPM sections involve 2x2 matrices. The model solves these by simply computing a Differences pattern of variance for the top row, and then selecting the best answer for the bottom row. If no answer scores above a criterion, the model attempts one strategy change: looking down columns, instead of across rows, to solve the problem.

### Evaluation

We evaluated our model by running it on sections B-E of the Standard Progressive Matrices test, for a total of 48 problems. Only section A was not attempted, as this section relies more on basic perceptual ability and less on analogical reasoning. While section B uses 2x2 matrices, sections C-E use 3x3 matrices of increasing difficulty.

Each problem from the test was recreated in PowerPoint and then imported into CogSketch. The experimenters segmented images into objects based on the Gestalt grouping principles (Palmer & Rock, 1994).<sup>3</sup> Recall that the model reorganizes the images into new sets of objects as necessary to solve a problem.

### Results

Overall, the model correctly solved 44/48 problems. To compare this level of performance to people, we converted this score to a 56/60 on the overall test, as individuals who performed this well on the later sections typically got a 12/12 on section A (Raven et al., 2000, Table SPM2). A score of 56/60 is in the 75<sup>th</sup> percentile for American adults, according to the 1993 norms (Table SPM13).

If our model captures the way people perform the test, then problems that are hard for the model should also be hard for people. The four missed problems were among the six hardest problems for human participants, according to 1993 norms (Raven, et al., 2000, Table RS3C3).

---

<sup>3</sup> In one problem, a dotted line was replaced with a gray line for simplicity.

## Discussion

Overall, our model matched the performance of above-average American adults on the Standard Progressive Matrices, both in the problems that it got right and the problems that it missed. Thus, it demonstrates that qualitative representations and the Structure-Mapping Engine can be used to model the performance of typical participants on this task. Importantly, structure mapping played a ubiquitous role in the model; it was used to compare objects, images, and patterns of variance. Additionally, these comparisons were used to rate similarities, identify differences, find corresponding elements, and produce generalizations. Thus, the simulation demonstrates that a single mechanism can be used to perform all the necessary comparisons in this complex task.

Direct comparison with BETTERAVEN (Carpenter, Just, & Shell, 1990) is impossible, as it was only built for, and run on, the Advanced Progressive Matrices. However, if we apply the principles of the model and assume perfect performance, it would achieve a 59/60, missing one of the problems missed by our model. Of the other three problems our model missed, two were due to insufficiencies in its representations of object and group attributes. Because it computes its own representations, our model provides a reason that these problems are more difficult for people, i.e. they require encoding more advanced attributes. Thus, while our model might solve fewer problems, its failures predict and explain human performance.

## Future Work

We have shown that our approach is sufficient for modeling human performance on Raven's Progressive Matrices. An important further step is to use the model to make new discoveries about how people perform spatial problem-solving. In a previous study (Lovett & Forbus, 2009), we used a similar model to identify possible cultural differences in the ways people represent space. RPM provides a number of unique opportunities to look at both spatial representation and analogical comparison, due to the complexity and diversity of the problems. By classifying problems based on the model strategies and model components required to solve them, we hope to gain a better understanding of both the factors that make one problem harder than another, and the cognitive abilities that make one person better than another at spatial problem-solving.

## Acknowledgments

This work was supported by NSF SLC Grant SBE-0541957, the Spatial Intelligence and Learning Center (SILC).

## References

- Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94, 115-147.
- Carpenter, P. A., Just, M. A., & Shell, P. (1990). What one intelligence test measures: A theoretical account of the processing in the Raven Progressive Matrices test. *Psychological Review*, 97(3), 404-431.
- Embretson, S. E. (1998). A cognitive design system approach to generating valid tests: Application to abstract reasoning. *Psychological Methods*, 3(3), 380-396.
- Falkenhainer, B., Forbus, K., & Gentner, D. (1989). The structure mapping engine: Algorithm and examples. *Artificial Intelligence*, 41, 1-63.
- Forbus, K., Nielsen, P., and Faltings, B. (1991). Qualitative spatial reasoning: The CLOCK project. *Artificial Intelligence*, 51(1-3), 417-471.
- Forbus, K., Usher, J., Lovett, A., Lockwood, K., & Wetzel, J. (2008). CogSketch: Open-domain sketch understanding for cognitive science research and for education. *Proceedings of the Fifth Eurographics Workshop on Sketch-Based Interfaces and Modeling*.
- Kuehne, S., Forbus, K., Gentner, D., & Quinn, B. (2000). SEQL: Category learning as progressive abstraction using structure mapping. *Proceedings of the 22<sup>nd</sup> Annual Conference of the Cognitive Science Society*.
- Lovett, A., Lockwood, K., & Forbus, K. (2008). Modeling cross-cultural performance on the visual oddity task. *Proceedings of Spatial Cognition 2008*.
- Lovett, A. Forbus, K., & Usher, J. (2007). Analogy with qualitative spatial representations can simulate solving Raven's Progressive Matrices. *Proceedings of the 29<sup>th</sup> Annual Conference of the Cognitive Science Society*.
- Lovett, A., Gentner, D., Forbus, K., & Sagi, E. (2009a). Using analogical mapping to simulate time-course phenomena in perceptual similarity. *Cognitive Systems Research* 10(3): Special Issue on Analogies - Integrating Cognitive Abilities, 216-228.
- Lovett, A., Tomai, E., Forbus, K., & Usher, J. (2009b). Solving geometric analogy problems through two-stage analogical mapping. *Cognitive Science* 33(7), 1192-1231.
- Markman, A. B., & Gentner, D. (1996). Commonalities and differences in similarity comparisons. *Memory & Cognition*, 24(2), 235-249.
- Palmer, S. E. (1977). Hierarchical structure in perceptual representation. *Cognitive Psychology*, 9, 441-474.
- Palmer, S., and Rock, I. 1994. Rethinking Perceptual Organization: The Role of Uniform Connectedness. *Psychonomic Bulletin & Review* 1(1): 29-55.
- Raven, J., Raven, J. C., & Court, J. H. (2000). *Manual for Raven's Progressive Matrices and Vocabulary Scales*. Oxford: Oxford Psychologists Press.
- Shepard, R. N., & Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science*, 171, 701-703.
- Snow, R.E., & Lohman, D.F. (1989). Implications of cognitive psychology for educational measurement. In R. Linn (Ed.), *Educational Measurement*, 3rd ed. (pp. 263-331). New York, NY: Macmillan.
- Vodegel-Matzen, L. B. L., van der Molen, M. W., & Dudink, A. C. M. (1994). Error analysis of Raven test performance. *Personality and Individual Differences*, 16(3), 433-445.