Cognitive Science 33 (2009) 1192–1231

ISSN: 0364-0213 print / 1551-6709 online
DOI: 10.1111/j.1551-6709.2009.01052.x

Solving Geometric Analogy Problems Through Two-Stage Analogical Mapping

Andrew Lovett, Emmett Tomai, Kenneth Forbus, Jeffrey Usher

Qualitative Reasoning Group, Northwestern University

Received 9 November 2007; received in revised form 4 May 2009; accepted 5 May 2009

Abstract

Evans' 1968 ANALOGY system was the first computer model of analogy. This paper demonstrates that the structure mapping model of analogy, when combined with high-level visual processing and qualitative representations, can solve the same kinds of geometric analogy problems as were solved by ANALOGY. Importantly, the bulk of the computations are not particular to the model of this task but are general purpose: We use our existing sketch understanding system, CogSketch, to compute visual structure that is used by our existing analogical matcher, Structure Mapping Engine (SME). We show how SME can be used to facilitate high-level visual matching, proposing a role for structural alignment in mental rotation. We show how *second-order analogies* over differences computed via analogies between pictures provide a more elegant model of the geometric analogy task. We compare our model against human data on a set of problems, showing that the model aligns well with both the answers chosen by people and the reaction times required to choose the answers.

Keywords: Analogy; Geometric analogy; Symbolic computational modeling

1. Introduction

One of the deep problems of cognitive science is how we make sense of the world around us. Humans have a powerful visual system, and it appears that part of its job is to compute descriptions of visual structure which can be used for recognition and understanding (Marr, 1983; Palmer, 1999). We have argued that qualitative spatial representations play important roles in medium- and high-level visual processing (Forbus, Ferguson, & Usher, 2001). Qualitative spatial representations provide a bridge between vision and cognition, as they seem to be computed by visual processes but take functional constraints into account. One way

Correspondence should be sent to Andrew Lovett, EECS Department, Northwestern University, 2133 Sheridan Road, Evanston, IL 60208. E-mail: andrew@cs.northwestern.edu

that spatial representations are used is in comparison tasks, for example, the geometric portion of the classic Miller Analogy Test. These problems have the form ''A is to B as C is to __?,'' as illustrated in Fig. 1. Such problems were used by Evans (1968) in the first computer model of analogy, his ANALOGY system. Evans' work was groundbreaking in that it showed that analogical problem solving could be modeled computationally.

This paper revisits this task, but with a new model, based on progress since then in modeling both visual processing and analogical processing. For visual processing, we use our work on sketch understanding (Forbus & Usher, 2002; Forbus, Tomai, & Usher, 2003). CogSketch (Forbus, Usher, Lovett, Lockwood, & Wetzel, 2008) is a computational system that constructs representations from human-drawn sketches and other line drawings. It represents a useful approach for modeling the understanding of visual structure because starting with digital ink allows the model to focus on processes of perceptual organization and ignore low-level issues such as edge detection. For analogical processing, we use the Structure Mapping Engine (SME) (Falkenhainer, Forbus, and Gentner, 1989; Forbus, Ferguson, & Gentner, 1994), a computational model based on Gentner's (1983) structure-mapping theory of analogy. We combine these models, along with two new ideas, to create a general-purpose model for geometric analogies. The new ideas are as follows:

1. *Structural alignment in mental rotation*. Analogical mapping provides an efficient technique for identifying the corresponding components of two spatial structures. We propose that structural alignment is used over qualitative shape representations to find the corresponding edges and determine what direction to start a more analog rotation-and-compare process that ascertains whether two shapes are rotations or reflections of one another.
2. *Second-order analogies*. Analogical matches highlight differences as well as commonalities (Markman & Gentner, 1996). Once structural alignment has been used to identify differences between pairs of figures, these differences can themselves be

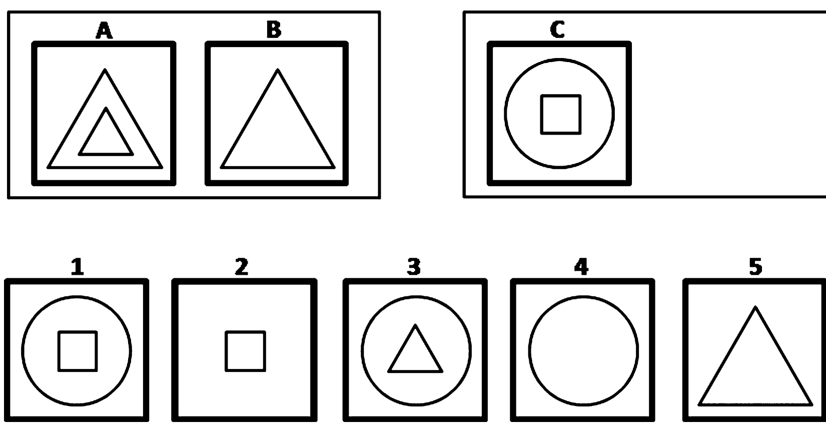


Fig. 1. A typical geometric analogy problem.

compared to determine how the mappings between two pairs of figures relate to each other.

These ideas, together with our models of structure mapping and visual sketch understanding, enable us to model the process of solving geometric analogy problems in its entirety, from encoding the input to selecting an answer.

Our model improves upon the original Evans model in two key ways. First, because CogSketch and SME have been used in a number of other tasks, our model is less problem-specific and better able to generalize to other tasks in visual perception and comparison. Second, because these components model human cognitive processes, they can be combined to create a task model that closely aligns with the way we believe people solve this task. In particular, our model is based on three core claims about human spatial reasoning and problem solving:

1. In encoding visual scenes for reasoning, people focus on the qualitative spatial relations between objects in those scenes.
2. Individuals compare visual representations using structure mapping, to identify both commonalities and differences.
3. Structure mapping plays a ubiquitous role in spatial problem solving, operating at the levels of comparing concrete physical shapes, finding analogical mappings between larger visual scenes, and performing more abstract second-order analogies between analogical mappings.

In order to evaluate our system as a model of human performance on geometric analogies, we compare its performance to that of human participants, looking at both the answers chosen and the time required to select an answer.

We begin the paper by briefly reviewing our analogy and sketch understanding models in Sections 2 and 3, highlighting the properties that play crucial roles in this higher-level task model. Section 4 describes our model of shape representation and comparison, showing how qualitative representations and structural alignment facilitate mental rotation. Section 5 describes how these components come together to model performance on the geometric analogy problems, including the process of two-stage structure mapping to compute second-order analogies. Section 6 compares our model with Evans' ANALOGY, and Section 7 shows that the model matches human performance on the 20 problems that Evans originally used (shown in Appendix Figs. A1–A7). Finally, Section 8 discusses other related work, while Section 9 draws some conclusions and describes future work.

2. Modeling analogy as structure mapping

According to the structure-mapping theory of analogy, humans compare two cases by aligning the common structure in their representations (Gentner, 1983, 1989). Structure consists of individual entities, attributes of entities, and relations among entities. There can also be higher-order relations between relations. A key point of structure mapping is that, when people align the structure in two representations, they show a systematicity bias. That is, they prefer to align deeper structure that involves higher-order relations. For example, consider a comparison between soccer and basketball. A match between the balls used in the games would not be supported by their color attributes, as the two balls are different colors. However, it would be supported by deeper structural commonalities. In each sport, a player moves a ball into a net in order to score points. The act of moving the ball through a net is a low-order relation common to the two cases. The causal relationship between this action and scoring points is a higher-order relation common to the cases. There is deep structural support for aligning the two balls, and so it is likely that a human would align them when comparing the two sports.

The Structure Mapping Engine (Falkenhainer et al., 1986; Forbus & Oblinger, 1990; Forbus et al., 1994) is a computer implementation of the comparison process of structure-mapping theory. Given two cases, a base and a target, it produces one to three *mappings* between the cases.[1] SME attempts to find mappings that maximize systematicity, the amount of structural overlap between the cases. A mapping includes a set of *correspondences* between the entities, attributes, and relations in the two cases; a list of *candidate inferences;* and a *structural evaluation score.*

Candidate inferences represent potential new knowledge about the target case that has been calculated from the base case and the mapping. When an expression in the base does not correspond to anything in the target, but the expression is connected to structure in the base that does correspond to structure in the target, a candidate inference is constructed. For example, suppose SME were fed the soccer and basketball cases, with soccer as the base case, and with the causal relationship left out of the basketball case. Based on the correspondences between moving the ball through the net and between scoring points in the two cases, SME would generate a candidate inference saying that because there is a causal relationship between these in the soccer case, there is likely also a causal relationship between them in the basketball case. Traditionally, SME only constructed candidate inferences from base to target. However, the process is symmetric, and in this paper we exploit SME's ability to construct *reverse candidate inferences*, that is, candidate inferences from the target to the base.

Finally, the structural evaluation score is an estimate of the similarity between the cases being compared, based on the systematicity of the mapping. It is calculated by assigning an initial score to each correspondence and then allowing scores for correspondences between relations to trickle down to the correspondences between their arguments. These local scores are used to guide the process of constructing mappings, so that mappings are more likely to contain deeper structures. For simplicity, we refer to the structural evaluation score as the *similarity score* for the remainder of this paper.

There is evidence suggesting that processes governed by the laws of structure mapping are ubiquitous in human cognition (Gentner & Markman, 1997). SME has been tested on a wide range of stimuli, ranging from concrete perceptual data (e.g., Forbus, Tomai, & Usher, 2003) to stories (Gentner, Rattermann, & Forbus, 1993) to abstract conceptual ideas (e.g., Gentner et al., 1997). This makes SME a natural choice for modeling comparison at multiple levels in the geometric analogy task.

3. Sketch understanding

We now turn to the problem of building representations for comparison. Our model automatically constructs spatial representations using CogSketch (Forbus et al., 2008), a sketch understanding system. CogSketch is based on the insight that when humans communicate through sketches, they do not assume that another person will recognize everything they sketch. Rather, they describe what they are sketching verbally, during the sketching process. Similarly, CogSketch lets its users tell it what they are sketching via a specialized interface. Thus, whereas most sketch understanding systems limit users to a single domain and focus on recognizing entities drawn from that domain, for example, electronic circuits, CogSketch is an *open-domain* sketch understanding system. Because its model of visual processing is general purpose and does not depend on recognizing entities drawn the user, it is not limited to a particular set of entity types prescribed in advance.

Every sketch drawn in CogSketch is made up *glyphs*. A glyph is a shape or object that has been drawn by the user. Each glyph is represented internally as *ink* and *content*. The ink consists of a set of *polylines*, lists of points describing lines drawn by the user when creating the glyph. The content is a conceptual entity representing what the glyph depicts. The user can describe the content by assigning conceptual labels to it from CogSketch's knowledge base. Each conceptual label is a term referring to some collection of entities. For example, House-Modern is the collection of typically sized houses.

CogSketch provides two mechanisms for creating glyphs. First, the user can draw each glyph, using CogSketch's start/stop buttons to indicate when they have stopped drawing one glyph and begun drawing a new one. Second, users can take shapes drawn in Microsoft PowerPoint and directly import them into CogSketch using copy-and-paste. This second method can be particularly useful for modeling purposes because psychological stimuli are often drawn in PowerPoint. By importing PowerPoint shapes, the user can evaluate CogSketch and connected models on the same stimuli that have been drawn in PowerPoint and given to human participants.

CogSketch also provides two mechanisms for segmenting a sketch into smaller sets of glyphs. First, glyphs can be drawn on different layers. Layers lie on top of each other visually, so glyphs that are drawn on different layers will appear to occupy the same visual space. However, CogSketch will only infer spatial relations between glyphs that have been drawn on the same layer. Second, glyphs can be entered into a grid called a *sketch lattice*. Glyphs that occupy different cells of the grid are considered spatially distinct, so CogSketch will only infer spatial relations between glyphs in the same cell of the sketch lattice. Fig. 2 gives an example of how sketch lattices can be used to segment a sketch into the separate pictures of a geometric analogy problem.

CogSketch automatically infers a number of spatial relationships among glyphs that have been sketched together. These include their relative positions, and the Region Connection Calculus (RCC8) relations (Cohn, 1996; see Fig. 3), a set of topological relations that describe whether two glyphs intersect or overlap, or whether one located inside another. Importantly, all of the relations computed by CogSketch are qualitative. For example, a

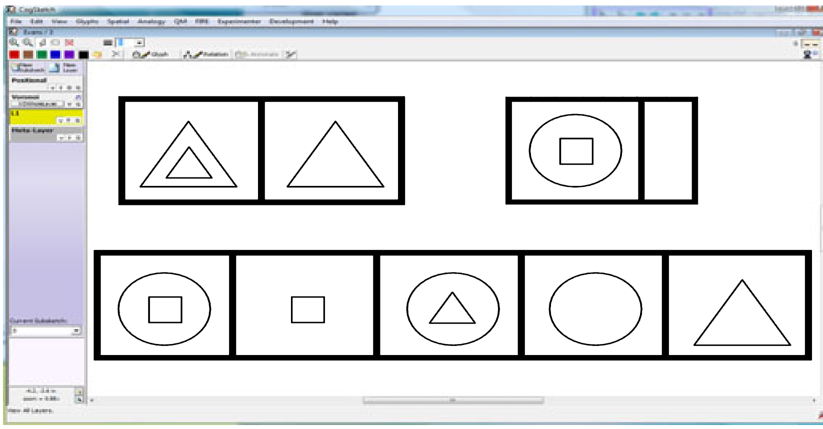


Fig. 2. A geometric analogy problem entered into CogSketch.

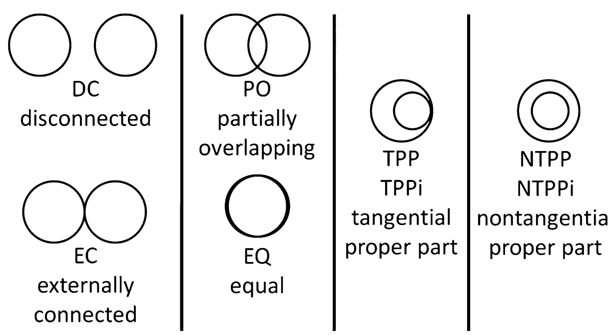


Fig. 3. The Region Connection Calculus (RCC8) relations for describing topology. The TPP and NTPP relations each have inverse relations, TPPi and NTPPi.

relative position would be *right-of* or *above*. Quantitative measures, such as the actual distance between two glyphs, are never used. We are committed to qualitative representations both because we believe they play a significant role in human cognition (Forbus et al., 2001), and because they are useful for analogical comparison. For example, consider a base sketch in which one glyph is 80 pixels right of another, and a target sketch in which one glyph is 100 pixels right of another. It is far easier to align the glyphs in these two sketches if one represents each sketch as having a glyph *right-of* another than if one represents the actual distances.

CogSketch combines the automatically inferred spatial information with the conceptual information supplied by the user to construct an overall representation of a sketch. This representation can then be used as the input for spatial reasoning tasks. In the simplest case, two sketches can be compared by feeding their representations into SME. More complicated tasks that have been performed using CogSketch and its predecessors include everyday physical reasoning problems (Klenk, Forbus, Tomai, Kim, & Kyckelhahn, 2005) and a visual oddity task (Lovett, Lockwood, & Forbus, 2008).

4. Shape comparison

At this point, it will be useful to refer to specific geometric analogy problems for motivation. The 20 problems used by Evans are given in Figs. A1–A7. For the remainder of the paper, we refer to each problem by its number. In addition, we refer to specific pictures in the problems. Pictures A, B, and C are the top three pictures in each problem. Pictures 1–5 are the bottom five pictures. These are the five possible answers to each problem.

One limitation of the spatial representations computed by CogSketch is that they do not give any consideration to the form of each glyph. They are simply based on where the glyphs are located in the sketch. However, a glyph's form plays an important role in solving geometric analogy problems. First, finding an analogical mapping between two pictures requires identifying the shapes in each picture. In almost every case, a square-shaped glyph will correspond to another square, whereas a circle will correspond to another circle. Second, it is sometimes necessary to compare two glyphs' forms and identify a transformation between them. Transformations include scaling, rotations, and reflections. Among the Evans' problems, Problem 3 requires distinguishing between a big triangle and a small triangle, Problem 6 requires recognizing that one shape is a rotation of another, and Problem 18 requires recognizing that one shape is a reflection of another.

In this section, we describe an extension to CogSketch that utilizes psychologically motivated representations of shape and a model of shape comparison that (a) can identify equivalent shapes across transformations, and (b) can determine the particular transformation that has been applied. The model is based upon the claim that the structural alignment of qualitative representations plays a key role in people's visual comparisons of shapes.

As the literature on mental rotation is particularly relevant, we begin by reviewing this literature. We then propose a simple model for how people accomplish mental rotation, arguing for the importance of structural alignment and qualitative representations. In Section 4.2, we describe our shape comparison model in detail.

4.1. Mental rotation

In mental rotation tasks, a participant is shown a geometric shape (often drawn in three dimensions) and then shown a second shape and asked to determine whether a rotation of the first shape would produce the second shape. In the first mental rotation experiments, Shepard and Metzler (1971) found that the amount of time required by participants in these tasks was proportional to the degrees of rotation between the shapes. This finding was significant because it suggested that humans were performing some kind of analog processing, rather than simply manipulating discrete symbols. However, the results naturally led to a second question. In three dimensions, there is an infinite number of paths that can lead between one orientation and another. Even in two dimensions, there are two possible directions of rotation around the center of a shape. Which path would humans take, and why would they choose it?

A number of studies have suggested that humans generally use the most efficient path between two orientations (e.g., Shepard & Cooper, 1982). The question of what is the most efficient path can become complicated in three dimensions, but for two dimensions this simply means that humans will mentally rotate a shape clockwise or counter-clockwise depending on which rotation will bring the mental image to the second shape's orientation more quickly. There has been some evidence that humans will not follow the most efficient path when it involves a particularly unfamiliar axis (Parsons, 1995), but in two dimensions there is only one axis of rotation. How then do humans know what path of rotation to use when rotating one object to align it with a second object, before they even know whether the two objects match?

Shepard and Cooper (1982) suggested that before mental rotating one of the representations, participants identify corresponding parts on the two shapes. These corresponding parts then guide the rotation. If this is true, then participants must have access to some rotation-invariant representation that can be used to identify the corresponding parts. Evidence for such a representation can be found in research on object recognition and object representation. Biederman and colleagues (Biederman, 1987; Biederman & Gerhardstein, 1993) have argued that people use a rotation-invariant, qualitative representation of objects for recognition. Their theory predicts that individuals should be able to recognize that two objects are from the same category equally well regardless of the different orientations of the objects, provided the objects are not seen from an unusual viewpoint that blocks a vital feature. In support of this theory, they have demonstrated that, under the right circumstances, participants are able to identify two instances of the same object at about the same speed regardless of the difference between the two objects' orientations.

Biederman's work has not gone unchallenged. Tarr, Bülthoff, Zabinski, and Blanz (1997) believe people use an orientation-specific representation for objects. They believe that when two objects at different orientations are being compared, people will usually need to mentally rotate one of the object's representations to align it with the orientation of the other representation. In support of their theory, they have demonstrated cases where the amount of time required to determine that two objects are the same is proportional to the degrees of rotation between the two objects. Such a result seems particularly easy to achieve when participants are working with unfamiliar, abstract shapes. This is unsurprising, as these are the types of shapes used by Shepard in the original mental rotation experiments.

We believe the best explanation for the results described above is that there are two separate representations used by humans in comparing visual stimuli. The first is an abstract, qualitative representation that describes how the parts of an object relate to each other. The second is a concrete, detailed, quantitative representation. The qualitative representation is rotation invariant; for example, opposite lines in a rectangle will be of approximately equal length and orientation regardless of the rectangle's orientation in the viewing plane. The quantitative representation, in contrast, is orientation dependent. For tasks that only require a quick, approximate comparison, individuals can simply compare two objects' qualitative representations. For tasks that require a more exact comparison, individuals must utilize the quantitative representations. To do so, they first rotate one of the objects' quantitative representations to align it with the other object's representation.

In mentally rotating a quantitative representation, individuals can be guided by the qualitative representations. We propose that they begin by using structure mapping on the qualitative representations to align the parts of the two objects. For example, if the objects are two-dimensional shapes, individuals may align the edges of the two shapes. We assume that this initial comparison is accomplished in about constant time. The mapping between the shapes is then used to guide the rotation. By comparing a single pair of corresponding parts, such as a pair of edges, one can quickly determine the rotational difference between them, as well as the shortest axis of rotation. The participant then must mentally rotate the remaining parts in the representation along that same axis of rotation. Because the parts must be rotated together along a single axis, this stage of the process can only be completed by most individuals in a deliberate, analog fashion, thus giving rise to the element of the reaction time that is proportionate to the degrees of rotation.

4.2. Modeling shape comparison

We model shape comparison with a four-step process (see Fig. 4 for the algorithm for shape rotations). In Step 1, each glyph's ink is decomposed into a set of edges. These edges represent the atomic parts of our shape representations. In Step 2, a shape representation is constructed to describe the qualitative relations between the edges in each glyph. In Step 3, the two glyphs' shape representations are compared using SME, in order to find the corresponding edges in the two glyphs. In Step 4, these correspondences are used to identify rotations, reflections, and changes in scale between the shapes.

In the following sections, we describe each step in detail. We focus on identifying rotations between shapes and give a brief description of how this approach can be modified to identify reflections and changes in scale.

```
1: Shape Decomposition
   Decompose each shape into its edges (see Fig. 5)
   Identify each shape's bounding edge set

2: Shape Representation
   For every edge in the bounding edge set
      Compute that edge's attributes
      Compute that edge's relations with other edges in the bounding edge set

3: Finding Corresponding Edges
   Compare shape representations with SME
   Use mapping computed by SME to identify corresponding pairs of edges

4: Comparing Corresponding Edges
   For each corresponding pair of edges
      Compute the difference Δ in their edge orientations
   If Δs are reasonably consistent across all edge pairs
      Return average (Δs)
```

Fig. 4. Algorithm for finding a rotation between two shapes.

4.2.1. Step 1: Shape decomposition

Decomposition of a glyph into its edges is accomplished via the Perceptual Sketchpad, an experimental extension to CogSketch. This system takes the polylines from CogSketch as its input. Each polyline is a list of points representing a line drawn by the user. While the polylines may correspond to the actual edges in a shape, as when a user draws a square as four separate lines, there is no guarantee that this will be the case. The user may draw an entire shape as one line, or the user may draw multiple lines for each edge. Therefore, the system makes no starting assumptions about whether the endpoints of the polylines represent endpoints of actual edges in the sketch. Polylines whose endpoints are adjacent are joined together, so that our starting input is a set of maximal lists of unambiguously connected points.

In traditional vision terms, the polylines may be thought of as the outline returned by an edge detector. An advantage of working with sketches is that we can explore problems in midlevel visual processing without needing to deal with the difficult first step of edge detection.

Edges are identified via a five-step algorithm (see Fig. 5). The algorithm begins with the naïve hypothesis that the polylines themselves are the edges. As it iterates over each step, the list of edges is progressively refined to reflect the actual edges of the shape. Note that while we have found the algorithm to be sufficient for performing the task of shape decomposition in this and other studies, we do not view it as a cognitive model; we believe further work is required to determine how people identify the parts of an object.

```
1: Connect Edges [3 scales]
   For every pair of edges
      If (distance between edge endpoints < α)
         Connect edges

2: Join Edges [2 scales]
   For every pair of connected edges
      If (no additional edges connected at the same endpoint)
         Join edges

3: Segment Edges [2 scales]
   For every edge
      Identify corners within the edge, based on discontinuities in curvature
      Segment the edge into multiple edges connected at the corners

4: Find T-Junctions [2 scales]
   For every pair of edges
      If (one edge's endpoint bisects the other edge, with a max distance
      between the endpoint and the edge of α)
         Segment the bisected edge into two connected edges
         Connect the bisecting edge to the two new edges

5: Find X-Junctions [1 scale]
   For every pair of edges
      If (the edges intersect, forming an X-junction)
         Segment each edge into two edges at the point of intersection
         Connect all four edges
```

Fig. 5. Algorithm for finding the edges of a glyph, based on incrementally refining the edge representation. The steps are repeated in order at up to three different scales (i.e., different values for α). However, some steps are only run at the first or the first two scales. The figure shows the number of scales at which each step is run.

In Step 1 of the algorithm, the system looks for connections between edges' endpoints. Two edges are considered to be connected if their endpoints lie sufficiently close to each other, as based on a distance threshold. A connection indicates that either (a) there is a corner between the two edges, as in two adjacent corners of a square, or (b) the two edges are actually part of a single edge.

In Step 2, pairs of unambiguously connected edges are joined to form single edges. Two edges are unambiguously connected if the junction between them contains only those two edges; that is, it is not a junction containing three or more edges. In this way, the algorithm creates maximally long edges between each junction of three or more edges.

In Step 3, each edge is segmented into a list of connected edges based on identifying points within the edge where there is a significant change in orientation. In this way, corners between edges, like the four corners of a square, are identified. Given hand-drawn sketches, it is easy to confuse random noise with actual corners. Therefore, three pieces of evidence are required to recognize a corner at a particular point:

1. A high curvature at the point, indicating that the orientation is changing sharply.
2. A high derivative of the curvature at the point, indicating that there is actually a discontinuity in the orientation, as opposed to a sharply but smoothly changing orientation, as would be found along a tight curve.
3. A difference in orientation between the part of the edge that precedes the point and the part of the edge that follows the point, indicating that the discontinuity in orientation is not purely local, as might result from noise.

Following Lowe (1989), we parameterize the list of points as two functions, f(x) and f(y), and convolve each function with the first and second derivative of the Gaussian to get the first and second derivatives of the x and y values at each point (dx, ddx, dy, ddy). The curvature at a given point can then be calculated via the function:

$$((\mathrm{dx} \times \mathrm{ddy}) - (\mathrm{dy} \times \mathrm{ddx}))/(\mathrm{dx}^2 + \mathrm{dy}^2)^{1.5}$$

In Step 4, the algorithm identifies T-junctions, where one edge's endpoint bisects another edge. As in the first step, T-junctions are identified based on a distance threshold between one edge's endpoint and the other edge.

Finally, in Step 5, the algorithm identifies X-junctions, where two edges intersect. These will prove useful in the shape representation, described in the next section.

The algorithm described above is based on the simplistic assumption that a single distance threshold can be used to determine whenever two edges are close enough to be considered connected. In reality, the human visual system appears to work on multiple scales (Marr, 1982), so individuals would likely both recognize and draw sketches in which there are different distances between connected edges. In order to better handle this issue, the algorithm runs at three different scales. It begins at the smallest scale, looking for the most obvious connections between edges. Afterwards, it increases the scale—that is, the distance threshold required to recognize that two edges are connected—and runs again, looking for new connections between those edge endpoints that have not yet been connected to anything. It then repeats this a third time. However, as the distance threshold increases and the algorithm becomes more permissive, the set of steps of the algorithm that are actually run becomes smaller. See Fig. 5 for the algorithm steps that are run at one, two, or three different scales.

Once edges have been identified, the system traces over the graph of connected edges to find the glyph's *bounding edge set*. The bounding edge set is the list of edges that are exterior in that glyph, that is, the edges that form the boundary of that glyph's shape. If a glyph is a closed shape, such as a square, then the bounding edge set will consist of a single *edge cycle*, the cycle of edges that run along the shape's exterior. Edge cycles are represented as lists of edges, in clockwise order. They may contain any number of edges; a cycle with only one edge is a circle or ellipse. Any edges located inside the loop formed by a cycle are *interior edges* and will not be included in the bounding edge set.

Not all glyphs are drawn as simple closed shapes. A more complex shape might have multiple edge cycles in its bounding edge set. On the other hand, if a glyph consists of only a corner between two edges (e.g., see Evans' Problem 6, Fig. 10), then the bounding edge set will consist of those two edges, and it will not contain any edge cycles.

The representations built by this process are, we conjecture, reasonable approximations of those that people compute for the sorts of abstract shapes used in tests such as geometric analogies. The particular sequence of steps in computing them reflects how this model is implemented in CogSketch; as stated above, we do not take them as necessarily representing the sequence of computations used in human visual processing.

4.2.2. Step 2: Shape representation

In Step 2 (see again Fig. 4), the shape representations are constructed. For each bounding edge set, a structural representation is computed in which the entities are the edges and the attributes and relations are those given in Table 1. As Table 1 shows, each edge is classified as straight, curved, or elliptical. Each edge is assigned a length attribute, based on its length relative to the longest edge in the bounding edge set. Beyond this, straight edges can be classified as vertical or horizontal. Those edges that are either vertical or horizontal are also classified as axis aligned. This is done because there appears to be a bias to map

Table 1
Attributes and relations used in the shape representation

Attributes	Connection Relations	Intersection Relations	Orientation Relations
Edge type (straight/curved/ellipse)	Corner-between (convex/concave)	Intersecting[a]	Parallel[a]
Relative length (relative to longest edge)	Connected[a] (not in an edge cycle)	Bisects	Perpendicular[a] (connected edges only)
Orientation (vertical/horizontal) (straight edges only)			
Axis-aligned (straight edges only)			

Note: [a]A symmetric relation.

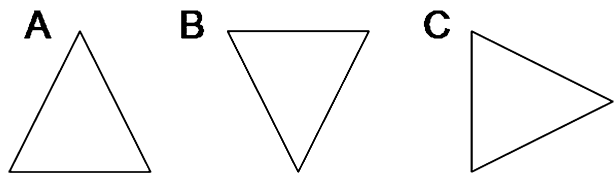


Fig. 6. Three triangles.

axis-aligned edges to each other when comparing shapes. For example, consider Fig. 6. Triangle B appears to be a copy of triangle A rotated 180°, whereas triangle C appears to have been rotated 90°. In each case, we believe there is a preference for mapping the axis-aligned edges to each other.

There are three types of relations between edges: *connection relations*, *intersection relations*, and *orientation relations*. The connection relations are corners between two adjacent edges in a cycle or general connections between two edges that lie outside of a cycle. Corners can further be classified as convex or concave. We have found that the convexity of an angle is a far more useful feature to include than the degree of an angle. Encoding the degree of an angle requires determining whether an angle is acute, right, or obtuse, but there can be a very fine line between, say, a right angle and an obtuse angle. In contrast, any time two edge's orientations are different enough to determine that there is a corner between them, no thresholding is required to say whether that corner is convex or concave.

The other types of relations are fairly straightforward. Intersection relations describe instances where two edges intersect at an X-junction, or one edge bisects another at a T-junction. Orientation relations describe when two edges are parallel or perpendicular. For large abstract shapes with many vertical and horizontal edges, the number of pairs of perpendicular or parallel edges can grow to an unmanageable size. Therefore, these relations are limited by only computing perpendicular relations between connected pairs of edges and only computing parallel relations between pairs of edges that both lie within the same edge cycle or both lie outside of any edge cycle.

4.2.3. Step 3: Finding corresponding edges

In Step 3 of the shape comparison, SME compares the two shape representations, returning one or more possible mappings between the edges of the two shapes.[2] A match between two irregular shapes such as the one from Evans' Problem 10 should produce only a single mapping. However, regular polygons, such as squares and equilateral triangles, allow as many mappings as there are edges. Of course, there is no guarantee that the mappings produced by SME are meaningful. Some or all of them may actually be incoherent, in the sense of not indicating any meaningful shape transformation. Thus, the next step is to evaluate the mappings.

4.2.4. Step 4: Comparing corresponding edges

The shape comparison process, as it is described up to this point, is based on our theory of how humans perform mental rotations. As in the cognitive theory described above, shapes are decomposed into edges, which are then aligned through structure mapping. In Step 4, our model diverges from that theory. Because we are working with a purely digital system, it does not actually rotate the shape in analog the way that we suspect a human would. Instead, it performs a series of mathematical comparisons. Each mapping is evaluated by iterating over one shape's bounding edge set and comparing the orientation of each of its edges to the orientation of the corresponding edge in the other shape's edge set. If the orientation difference for each edge pairing is about the same, the model concludes that the two shapes are rotations of each other. If not, the SME mapping is rejected.

There may be multiple valid mappings between two shapes, and thus multiple possible rotations between them, so our model ranks the mappings. As the work on mental rotations shows, humans generally find the shortest rotation between two shapes. Thus, we rank rotations found from fewest to most degrees of rotation. If two shapes are identical, this means the first rotation returned will be close to zero degrees of rotation. We concede that this method is not viable in psychological terms. Humans generally identify the shortest rotation first, rather than finding multiple rotations and then picking the shortest from among them. The model could be made more realistic by adding attributes and relations to the structural representation to bias SME toward giving the highest similarity score to the mapping that results in the smallest overall rotation. This might be done by including some orientation-specific facts in the representation. Then mappings could be ranked according to their score before evaluation.

4.2.5. Shape comparison for reflection

Identifying reflections between shapes requires two changes to the approach described above, one in the representation and one in the comparison. First, in the representations, the order of the edges in all edge cycles must be reversed for one shape. This is because any axial reflection of a two-dimensional image reverses the order of that image's edges.

Second, after SME has found a mapping between the two shapes' edges, the model evaluates the mapping by iterating over each corresponding pair of edges and identifying possible axes of reflection between those two edges. For example, if the base edge is horizontal and the target edge is vertical, the transformation between them could involve a reflection over a diagonal, 45° axis. As with rotations, there must be a consistent axis of reflection for all corresponding edge pairs for the mapping to be considered valid.

4.2.6. Shape comparison for changes in scale

Identifying a change in scale between two shapes depends on first recognizing that they are the same shape by finding a rotation or reflection between them. After the model has determined that the two glyphs are the same shape, it can find a change in scale by simply comparing the dimensions of the two glyphs' bounding boxes.

4.2.7. Special shape types

There are two special shape types that can be identified without the normal shape comparison process: ellipses and dots. Ellipses are closed shapes consisting of only a single, elliptical edge, that is, circles and ovals. Dots are very small glyphs with no real discernible shape. For both of these shape types, the model never looks for rotations and reflections, as their edges have no real orientation. In addition, the model never looks for changes in scale between dots.

5. Geometric analogy model

We now describe how the overall model performs geometric analogies using input from CogSketch. Fig. 7 shows the system architecture for the model. We will discuss each of the three components of the model, focusing particularly on how SME is used to perform two-stage analogical mapping.

5.1. CogSketch: Encoding representations

Geometric analogy problems are constructed in CogSketch using three sketch lattices (see Fig. 2). Each entry in a sketch lattice corresponds to a particular picture in the problem. CogSketch automatically constructs representations of the pictures via two steps: computing spatial relations between the glyphs in each picture, and computing shape relations between the glyphs across the entire problem.

The model makes use of two types of spatial relations computed by CogSketch: positional relations and topological relations. The positional relations used are *right-of* and *above*. The topological relations are a subset of the RCC8 relations (see Fig. 3) EC, PO, EQ, TPP, and NTPP. The *disconnected* (DC) relation is not used because it simply represents the lack of any topological relationship between two glyphs. Two other relations (TPPi, NTPPi) are not used because they are inverses of other relations and therefore redundant.

CogSketch performs shape comparisons across all the glyphs in the problem. Glyphs for which there is a valid rotation or reflection between them are grouped into *shape equivalence classes*, that is, groups of glyphs that are the same shape. In this way, CogSketch is able to determine which glyphs in the problem are the same shape without needing to know

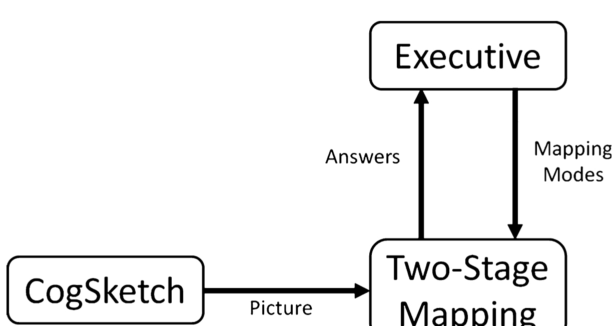


Fig. 7. System architecture.

the appropriate shape types ahead of time. Because CogSketch does not know the names of actual shape types, each shape equivalence class is assigned a *shape name*, an arbitrary symbol to represent it while the problem is being solved.

CogSketch also determines whether any glyphs are textured. At present, texture is identified by the simple heuristic of looking for closed shapes that contain straight, internal edges that intersect the shape's exterior edges, for example, a square with a pattern of vertical lines inside it. As with shapes, those glyphs with textures are grouped into texture equivalence classes based on the orientation of the internal edges, and each equivalence class is assigned a texture name.

CogSketch's output is a *picture representation* for each picture of the problem. The picture representation consists of the list of glyphs, the spatial relations between those glyphs, and the shape and texture equivalence classes for each glyph.

5.2. Two-stage mapping: Identifying an answer

As noted above, our model solves analogy problems through a two-stage analogical mapping process. Fig. 8 summarizes this process. In the first stage, the model compares two pictures in the problem, such as A and B. SME's mapping between the representations of these pictures includes a set of candidate inferences that describe the differences between the representations. These candidate inferences are used to construct a representation Δ(x,y) that describes how pictures x and y differ. These Δs are used, in turn, as the input to the second-stage analogy. In the second stage, the Δ(A,B) is compared to Δ(C,n), where *n* is

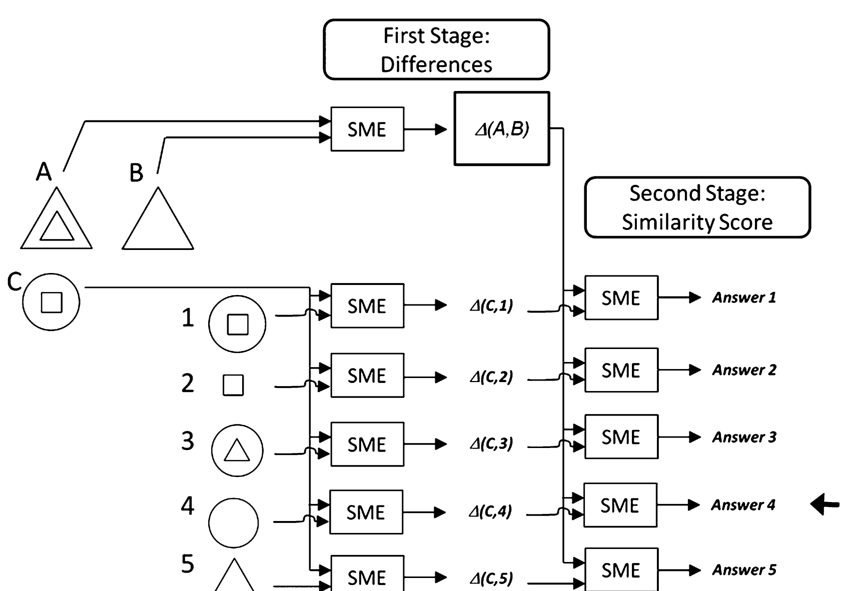


Fig. 8. Two-stage structure mapping.

each of the five possible answer pictures. SME's similarity score tells the model the strength of each second-stage mapping. The mapping computed via this second-order analogy process with the highest score is declared the winner.

Next we describe the representations used and the comparisons performed for each of the mapping stages in greater detail.

5.2.1. First-stage mapping

In the first stage, picture A is compared to picture B via SME to compute Δ(A,B), the set of differences between A and B. Similarly, C is compared to each of the five possible answers to compute Δ(C,n), the differences between C and that answer. The representations used for each comparison are computed from the picture representations produced by CogSketch.

For a given first-stage comparison, the model constructs a *comparison representation* for each of the two pictures being compared. This representation contains the spatial relations computed by CogSketch, the shape/texture attributes, and the shape relations. The shape and texture attributes are the names that have been assigned to each equivalence class, as described above. The names are kept consistent for each equivalence class throughout the problem. In this way, the model can recognize, for example, that both Δ(A,B) and Δ(C,2) involve the removal of a particular shape, a dot, in Evans' Problem 15.

Shape relations describe transformations between shapes that are in the same equivalence class, that is, rotations, reflections, and changes in scale. Unlike shape attributes, they are recomputed for each comparison representation, so as to capture only shape transformations that are occurring within the pair of pictures being compared. For each shape class, the model picks one object in the two pictures and designates it as the *reference shape*. When possible, the reference shape is chosen from the base picture (i.e., picture A or picture C). Then all other objects from the same shape class are compared to it.

Consider, for example, Evans' Problem 3. When pictures A and B are compared, all three glyphs are in the same shape class. Supposing the larger triangle in the base picture were chosen as the reference shape, the comparisons would lead to the conclusion that the pictures contain two regular-sized shapes and one smaller shape.

Identifying rotations and reflections between shapes can be tricky because when the shapes contain symmetries or regularities, there will be ambiguity. Consider Evans' Problem 12. There could be either a rotation or a reflection between the ''B'' shapes in pictures A and B. Deciding whether to encode the shape relation as a rotation or reflection is particularly important here because the answer to the problem depends on one's choice: If there is a rotation between them, the best answer is 1, but if there is a reflection, the best answer is 3.

The problem is further complicated by the fact that in some cases (e.g., the octagons in Problem 14), two shapes are identical, but because of symmetry there is still a possible rotation or reflection between them. Thus, it is necessary to rank these three possible shape relations (identity, rotation, reflection) by order of priority, so that one can be chosen when there is ambiguity. It seems clear that if the shapes are identical, this will be particularly salient, regardless of other possible transformations. Therefore, the model ranks identity first. After this, the model ranks reflection before rotation, meaning we are predicting that reflections will be more salient than rotations. This is primarily an intuitive guess, which we evaluated when we ran the set of problems on human participants (see below).

Each glyph is assigned an attribute designating its rotation (including an attribute for ''no rotation''), its reflection, and its size relative to the reference glyph. The full set of terms used in the comparison representations is given in Table 2. Once the comparison representations for the two pictures have been computed, these representations are compared via SME. SME computes an analogical mapping between the representations and produces candidate inferences both forward (from the base to the target) and in reverse (from the target to the base). These forward and reverse candidate inferences describe elements of the base and target representations that failed to align. The candidate inferences are used to compute the Δ's, the representations of the differences between the two pictures.

5.2.2. Second-stage mapping

The candidate inferences from the first stage, that is, the Δ's constituting the model's representation of the differences between two pictures, are used as the input to the second stage (see again Fig. 8). The model uses SME to align the Δ produced by mapping pictures A and B with the Δ's produced by mapping C to each of the five possible answers. Each answer *n* is scored on the structural strength of the second-stage mappings, that is, on SME's similarity score between Δ(A,B) and Δ(C,n).

The first-stage mappings are between concrete representations of visual stimuli. In these mappings, SME allows for only identical attribute matches. That is, the match between two entities is supported by shape if they are both circles, but not if one is a circle and one is a square. The second-stage mappings, in contrast, are between abstract sets of differences. It is no longer only particular shapes that matter, but changes between them as well. In order to capture the greater flexibility in the second-stage mappings, we construct more general predicates for most of the attributes and relations used in the first-stage mappings. Table 3 provides examples of how candidate inferences from the first-stage mappings are generalized. Note that the representations we use here are merely a best guess, based upon the constraints of the task. We believe there are still deep questions about how people generalize knowledge. We now describe our model's generalization process in detail.

In computing a particular Δ, each candidate inference or reverse candidate inference is transformed into a set of primary facts and supporting facts. The primary facts usually contain a generalization of the original inference. For example, there is a generalized form for each of the attribute types (shape, texture, rotation, reflection, shape-size). Each of these

Table 2
Attributes and relations used in the first-stage comparison representation

Attributes	Relations
Shape type	Positional (left-of, above)
Texture type	Topological (RCC8)
Rotation (includes rotation-none)	
Reflection (includes reflection-none)	
Size (RegSize, SmallSize, BigSize)	

Table 3.
Examples of how first-stage inferences from the target and base are generalized to create the second-stage representation (some predicates have been simplified for clarity). Supporting facts are also included in the second-stage representation but are structurally less important.

Expression Type	Inference from Base	Inference from Target	Primary Second-Stage Facts	Supporting Facts
Glyph Size Attribute	(SmallerElement O1)		(**candidateInference** (ScaledElement-Generic O1))	(SmallerElement O1)
Glyph Shape Attribute		(Shape Type0 O2)	(**reverseCandidateInference** (ShapeType-Generic O2))	(ShapeType0 O2)
Glyph Rotation Attribute	(ShapeRotated-90 O3)		(**candidateInference** (ShapeRotated-Generic O3))	(ShapeRotated-90 O3)
Positional Relation	(leftOf O4 O5)		(**candidateInference** (positional-Generic O4 O5))	(leftOf O4 O5)
Positional Relation (Reversal)[a]	(leftOf O6 O7)	(leftOf O8 O9)	(**inferenceReversal** (positional-Generic O6 O7))	(leftOf O6 O7) (leftOf O8 O9)
Topological Relation	(partiallyOverlapping O1 O2)		(**candidateInference** (partiallyOverlapping O1 O2))	—
Relation with Unmapped Shape	(above O3 (SkolemFn O4))		(**candidateInference** (positional-Generic O3 (SkolemFn O4))) (**candidateInference** (SpatialThingExists (SkolemFn O4)))	(above O3 (SkolemFn O4)) (RegSizeElement (SkolemFn O4)) (ShapeType3 (SkolemFn O4)) …etc…

Note: [a]InferenceReversal would only be used in this case if O6 corresponded to O9 and O7 corresponded to O8 in the first-stage mapping.

generalized forms captures the fact that, in the first-stage mapping, there has been some kind of change in shape, or some kind of rotation, etc., without specifying the particular shape, or the degrees of rotation. In this way, the second-stage mapping might align two Δ's because they both contain rotations, despite the fact that the rotations are not the same. There should, however, be a preference for exact matches. Thus, the supporting facts contain the more detailed information (the exact shape, the exact number of degrees for rotation, etc.). In order to give the primary facts primacy over the supporting facts, they are placed in a larger expression, usually of the form (candidateInference <primary fact>) or (reverseCandidateInference <primary fact>). Because of SME's systematicity bias, this additional layer of depth makes the primary facts much more important in the mapping. However, everything else being equal, a comparison in which the supporting facts also align will receive a higher similarity score.

As Table 3 shows, positional relations are also generalized to produce a generic positional relation.[3] This seemed necessary to capture the fact that two of Evans' problems (15, 20) involve mapping a change in horizontal position to a change in vertical position during the second-stage comparison. Whether this type of generalization is appropriate can be evaluated by looking at whether human participants have a particular difficulty with these problems; if they do not, then it is likely that they can easily generalize positional relations. Note that the model does not generalize the topological relations. It is unclear how or whether it would make sense to generalize such relations, so at present the model keeps the specific relation for its primary fact and produces no supporting facts.

In addition to candidateInference and reverseCandidateInference, the model produces one more type of primary fact: inferenceReversal (again, see Table 3). This captures the case in which the same relation is found in both the candidate inferences and the reverse candidate inferences, but the order of the arguments is reversed. For example, a reversal of a left-of would indicate the change between a square left of a circle and a circle left of a square, while a reversal of a topological containment relation would indicate the change between a square inside a circle and a circle inside a square. The inferenceReversal term seemed necessary again because of the flexibility required in reasoning about positional relations. Solving Evans' Problem 20 requires recognizing that there has been a reversal in a positional relation, though the relation is a vertical one for Δ(A,B) and a horizontal one for Δ(C,3).

Finally, there are some cases where a candidate inference contains an *analogy skolem*, which indicates that one of the objects in the base (for a forward inference) fails to map to anything in the target. For example, in Problem 15, picture A contains a dot and a square, while picture B contains only a dot. In this case, the candidate inference in Δ(A,B) would be a left-of relation in which one of the objects in the base (the dot) failed to map to anything in the target, and thus it would be represented as an analogy skolem (''SkolemFn'' in Table 3). When this occurs, two additional steps are taken. First, an extra primary fact is created to indicate the presence of an extra shape in the base or target, as this would seem to be one of the significant differences between the two pictures. Second, supporting facts for all of this extra shape's attributes are added. This lets SME align two Δ's based on the fact that they both involve, say, a dot disappearing, or a rotated object appearing.

Once the final Δ's are computed, SME compares Δ(A,B) to Δ(C,n) for each of the possible answers. Each answer is scored based on SME's similarity score. Scores are normalized based on the size of the base and target to avoid giving any answer an advantage simply because its Δ is larger. The answer whose second-stage mapping receives the highest similarity score, that is, the answer whose mapping to C is most similar to the mapping between A and B, is chosen as the correct answer.

5.3. Executive: Controlling the two-stage mapping process

The two-stage mapping process, as described above, is sufficient for answering most of the geometric analogy problems in the Evans set. However, it is based on the assumption that the initial mapping produced between any pair of pictures in the first stage is always correct. This is not necessarily the case. For example, in Problem 17, the initial mapping between A and B matches the big triangle in A to the big triangle in B, resulting in a Δ(A,B) in which the smaller triangle disappears. This Δ intuitively makes sense and clearly should be the model's first guess. However, this Δ(A,B) fails to match the Δ's between picture C and any of the five possible answers. In order to solve this problem, the model must build a set of first-stage mappings, look for an answer, and, when an answer cannot be found, backtrack to the first stage to look for new mappings. In this case, it must recognize that an alternate mapping between A and B involves the bigger triangle disappearing while the smaller triangle grows in size. This Δ(A,B) then maps to Δ(C,4), in which the bigger circle disappears while the square grows in size.

We deal with the problem of backtracking by utilizing an additional model component: the *executive* (see again Fig. 7). After the two-stage mapping component identifies a possible answer to the problem, the executive evaluates the answer to see whether it is a *sufficient* answer. An answer is sufficient if, in the second-stage mapping, all the primary facts in Δ(A,B) and Δ(C,answer) align. If some primary facts fail to align, the executive deems the answer insufficient and requires the two-stage mapping component to look for a better mapping.

New mappings are created through the use of four *mapping modes* for the first-stage mappings. In the Regular mode, the mapping behaves exactly as described earlier. In the Alternate mode, the model uses SME to find an alternate mapping between the two pictures. It requires that all of the glyphs in this mapping be aligned with different glyphs than they were aligned with in the original mapping.

There are two other modes: Reflection and Rotation. In these, the model changes the priorities regarding whether two shapes in the same shape class should be considered identical, reflections of each other, or rotations of each other. In Reflection mode, reflections receive the highest priority, whereas in Rotation mode, rotations receive the highest priority. Reflection mode is useful in Problem 18, where the ''B'' shapes in pictures C and 3 appear to be identical. Only when they are reinterpreted as reflections over the x axis does this Δ align with Δ(A,B). Similarly, the octagons in Problem 14 all appear to be identical, but when the octagons in C and 4 are reinterpreted as rotations of each other, they align with Δ(A,B).

Each of the non-Regular mapping modes requires looking at a problem in an unusual way that is different from the way we believe a person would normally approach the problem. To prevent them from being overapplied, there are strong constraints on when each mapping mode may be used. The Alternate mapping mode is only attempted if there is an alternate mapping that receives a similarity score close to the score SME gives to the best mapping. Any time a first-stage Alternate mapping is attempted, its score is compared to the best mapping's score. If their ratio falls below a threshold, the mapping is abandoned before any second-stage mappings are considered. For the present study, we found that a threshold of 60% allowed the model to rule out alternate mappings that fell far below the best mapping while considering mappings that approach the best mapping in structural strength.

The Reflection and Rotation mapping modes are particularly dangerous, as they allow identical shape matches to be reinterpreted as reflections or rotations. At present, the model is constrained to only consider them when the two pictures being compared each contain only a single glyph. Thus, the model will only consider a nonintuitive transformation between two shapes when those shapes are the only elements in the pictures being compared.

The executive searches for a sufficient answer by varying two parameters: the *base mapping mode*, that is, the mapping mode used when comparing pictures A and B; and the *target mapping mode*, the mapping mode used when comparing picture C to each possible answer. It begins with Regular for both mapping modes, searches over all pairings of Regular, Reflection, and Rotation, and finally considers those pairings that include Alternate mapping modes. Rather than always searching through all pairings, the executive concludes the search as soon as a base/target mapping mode pairing produces a sufficient answer. If no sufficient answer is found, it instead chooses the highest-scoring answer found across all mapping modes.

5.4. The question of shape identicality

The model of the geometric analogy process described above suggests that people will utilize the same creative, flexible comparison processes in geometric comparisons that they use in more abstract analogies. However, it is possible that the task of comparing geometric pictures encourages people to develop conservative biases that constrain the mapping process. In particular, we believe that people may exhibit a *shape identicality* bias, meaning that in the first-stage picture comparisons, they will only map between objects that are the same shape. Such a conservative bias would not be predicted by the present model because each object's shape is represented only as a single attribute. Thus, while the corresponding attributes encourage SME to map objects of the same shape to each other, it is quite easy for other parts of the structure to cause objects of different shapes to align.

We view the issue of shape identicality to be an open question. To test it, we have created two overall modes for the geometric analogy model. In the Normal mode, the model operates as described above. In the Shape Identicality mode, the model takes the more conservative approach to geometric analogies by utilizing *partition constraints* in SME's first-stage mappings. Partition constraints are an optional feature of SME in which the system can specify that entities of certain types can only map to other entities of the same types. In this case, the partition constraints are applied to the shape types, so that entities can only map to other entities of the same shape in the first-stage mappings. In Section 7, we compare both modes to human performance, in order to determine whether humans exhibit a shape identicality bias.

6. Comparison With ANALOGY

How does this model compare with Evans' (1968) original system? ANALOGY consisted of two parts. Part 1 built representations of each of the eight pictures within a given problem, and Part 2 selected an answer. The program was split into two parts in order to fit into the punch-card machine available at the time. We describe the parts in turn and then compare them to CogSketch and SME in our model.

ANALOGY's inputs were lists of points and vertices, although for 9 of the 20 examples, Evans hand-coded the representations instead. Part 1 computed shape descriptions for these lists of points, and spatial relationships between the shapes, such as INSIDE, LEFT, and ABOVE. Part 1 also compared pairs of shapes to identify rotations and reflections. The shape comparison was done by searching exhaustively over every correspondence between vertices in the two shapes with the same degree and between edges in the two shapes with the same curvature in order to find every possible transformation between the shapes.

Part 2 of ANALOGY solved problems through two stages of mapping, as in our own approach. However, the mapping process and the output representations were quite different. In the first stage, the system found all possible correspondences between the objects in the two pictures such that corresponding objects were always of the same shape. For example, if each picture contained one square and one circle, there would only be one possible mappings between them, in which the two squares were matched and the two circles were matched. If each picture contained three squares, there would be six possible mappings between them. Unmatched shapes in the base picture were classified as *shape removals*, while unmatched shapes in the target sketch were classified as *shape additions*. The output of this stage was a rule containing all shape matches, shape removals, and shape additions for a particular mapping. The rules also included all relations and attributes associated with each of the shapes. A separate rule was generated for each possible mapping, so potentially a large number of different rules might be generated when two pictures were being compared.

In the second stage, ANALOGY compared each rule generated between pictures A and B to each rule generated between picture C and each of the five answers. In this stage, the system looked for every possible mapping between the shape matches, shape removals, and shape additions in the two rules being compared. For each of these mappings, the system calculated the number of attributes and relations common to the corresponding components of the two rules, thus generating a measure of rule overlap. The system chose the answer whose associated rule overlapped the most with one of the A/B rules.

Our model differs from ANALOGY primarily in that it uses a single comparison process—SME—for three separate comparisons: comparisons between shapes, first-order comparisons between pictures, and second-order comparisons between the outputs from first-order comparisons.[4] In contrast, ANALOGY used a different process for each of the three types of comparison. As noted above, SME models what we believe to be a general cognitive process of structural alignment. Our model for the geometric analogy task provides additional evidence for this view, as it shows how the same process can play multiple roles even within a single task. Depending on the input that is provided, it can align concrete shapes, spatial relationships, or even abstract sets of differences between spatial relationships based on its own output. By showing that SME can be used in all these roles, we believe we have strengthened the argument for the importance of structure mapping in human cognition.

Another major advantage of SME is that it returns only a small number of mappings for each comparison. Note that ANALOGY performed an exhaustive search for each of the three types of comparisons. Thus, in shape comparison, the system identified every possible transformation between two shapes. In the first-order comparisons, the system returned every mapping that matched objects of the same shape. If two sketches each contained n_1

squares and n_2 circles, it would return ($n_1! \times n_2!$) mappings between them. In the second-order comparisons, the system identified every mapping between the shape matches, shape removals, and shape additions in the two rules being compared. If the two rules each contained n_1 matches, n_2 additions, and n_3 removals, the system would return ($n_1! \times n_2! \times n_3!$) mappings.

In contrast, SME uses the structure of the descriptions being compared to identify the best or the best few mappings. In the shape comparisons, the model performs one comparison looking for rotations and one comparison looking for reflections, so that it returns at most three values: an identity match, a minimal rotation, and a reflection. In the first-stage comparisons, the model finds at most four mappings, one for each mapping mode (although in most cases the first mapping with the Regular mode is sufficient). In the second-stage comparisons, SME computes only a single similarity value for each pair of Δ's.

In building the ANALOGY system, Evans demonstrated that a computer program could solve geometric analogy problems. We believe his results were extraordinary, given the technology of the time, and we would not suggest that Evans failed in any way to accomplish his goal. However, we built our own model with a different goal: to show that a computer simulation can solve these problems in the same way a human solves them. For this reason, it was important that our model both use components based on proposed cognitive processes and search through the problem space in the same way a human might.

7. Experimental study

In order to evaluate our computational model, we reconstructed the original 20 Evans problems in PowerPoint (see Appendix Figs. A1–A7) and gave them to human participants to assess their performance on the task. The same PowerPoint figures were imported into CogSketch, and the computational model was run on these figures. Thus, we were able to compare human performance and the model's performance on the same stimuli.

7.1. Methods

7.1.1. Behavioral study

The Evans problems were given to 34 human participants. Participants were given a description of the geometric analogy task followed by two simple geometric analogy example problems (without feedback) before they saw the 20 Evans problems. Both the ordering of the problems and the ordering of the five possible answers for each problem were randomized across participants.[5]

Before each problem, participants clicked on a fixation point in the center of the screen to indicate readiness for the next problem. After the problem was presented, participants clicked on the picture that they believed best completed the analogy. Participants were instructed to be as quick as possible without sacrificing accuracy. The two measures of interest for each problem were the answer chosen and the time taken to solve the problem.

7.1.2. Computer simulation

The 20 Evans problems were imported into CogSketch via copy-and-paste. Because each PowerPoint shape was converted into a single glyph, CogSketch was not required to segment the image into glyphs. Sketch lattices were used in CogSketch to manually segment the full set of glyphs into the individual pictures of the problem (A, B, C, and 1–5); see Fig. 2 for an example. CogSketch then automatically decomposed each glyph into its edges and constructed shape representations for each glyph and picture representations for each picture.

The model was run in two modes: Normal mode and Shape-Identicality mode. For each mode and each problem, the model reported the following:

1. The highest-scoring answer.
2. The best score for each of the five answers.
3. The first-stage mapping modes that were used to achieve these scores.
4. The number of SME first-stage and second-stage comparisons required to solve the problem.

This last measure can be seen as an approximation of the time required to solve the problem. All problems for which an answer can be found on the first two-stage mapping attempt will involve the same number of comparisons. Only those problems on which the model must explore alternative first-stage mapping modes to arrive at an answer will require more comparisons. We predicted that those problems requiring more comparisons would be the problems which required the most time for the human participants to complete.

7.2. Results

7.2.1. Behavioral results

Figs. A1–A7 present the percentage of people who chose each answer on a problem and the average reaction time on that problem. Overall, the results show a remarkable degree of consistency across participants. All participants chose the same answer for 9 of the 20 problems, while over 90% chose the same answer for seven additional problems. The greatest disagreement was on Problem 17, on which only 56% of participants chose the same answer, although this was still a statistically significant preference for one answer over the others ($p < .001$). Henceforth, we refer to the answer chosen by the majority of participants on a problem as the *preferred answer* for that problem.

The time required to solve a problem varied between 4.5 s and 26.7 s, with a mean of 9.6 s and a median of 7.0 s. Two problems, 10 and 17, required at least 10 s more than any of the other problems, suggesting qualitative differences in the steps that were required to solve these problems. We consider these problems further below.

One question of interest was whether participants would prefer rotation or reflection on those problems where the answer was ambiguous (Problems 12 and 19). The results supported our initial hypothesis: Participants preferred reflection, with 97% choosing the reflection-based answer on Problem 12 and 82% choosing the reflection-based answer on Problem 19.

Another question was whether participants would have difficulties on the problems that involved generalizing positional relations in order to find a mapping between a vertical positional relation and a horizontal positional relation in the second-stage comparison. Three problems (7, 15, and 20) involve this type of mapping. Across these problems, an average of 98% of participants chose the preferred answer. The average time required was 7.7 s, below the average and only slightly above the median for the 20 problems. Thus, participants did not appear to have had trouble generalizing positional relations.

7.2.2. *Simulation results*

Both modeling modes, Normal and Shape-Identicality, chose the preferred human answer on all 20 Evans problems. In order to further evaluate each model's ability to predict human performance, we considered the correlation between a model's number of comparisons required to solve a problem and the human timing data. The Normal mode had a .59 correlation with the human timing data, while the Shape-Identicality mode had a .75 correlation. Thus, while both modes show a strong correlation with the human data, the Shape-Identicality mode appears to model it more closely.

Looking at the individual problems, the main difference between the modes is on Problem 10. The Normal mode is able to easily solve this problem by stipulating that in the mappings between A and B and between C and 5, the circle changes to a dot, while the dot changes to the circle. In contrast, the Shape-Identicality mode maps the circle to the circle and the dot to the dot, and thus has more difficulty solving the problem; this mode still gets the correct answer on the first pass, but it does not deem the answer sufficient, so it tries additional passes with other mapping modes. Given that the human participants required 23.7 s to solve this problem, it seems most likely that their problem solving was constrained by shape identicality.

7.3. *Discussion*

Both modeling modes match the human data quite well. However, Shape-Identicality mode appears to provide a more accurate model of human performance. One conclusion we might draw from this is that, when individuals solve geometric analogy problems, they utilize a conservative bias in the mapping process, only allowing mappings between objects of the same shape. Another possibility, however, is that they perform two-stage structure-mapping as modeled in the Normal mode, but they utilize representations that give shape characteristics more weight. In the current model, shape is represented only as a single attribute. If shape played a greater role in the representations, then the Normal mode would exhibit the same behavior as the Shape-Identicality mode on all Evans problems.

We can further evaluate the model by looking at its ability to explain the second most popular choice on problems where individuals showed some disagreement. There were three problems—4, 17, and 19—where over 10% of participants chose some answer other than the preferred answer. We now consider each of these in turn, comparing the human results to the model in Shape-Identicality mode.

On Problem 4, 24% of participants chose Answer 2, rather than Answer 4. Our model successfully predicts that Answer 2 will be the second choice—while the model gives Answer 4 a perfect score of 1.0, Answer 2 received the second highest score at 0.95. Looking at the model's representations, Δ(A,B) and Δ(C,2) receive a high similarity score because they both involve a change in positional relations, with one object moving to the right of another object. Δ(C,2) does not score as high as Δ(C,4) because there is an additional positional relation in Δ(C,2): The dot moves to be above the other object. However, this additional positional relation has only a small effect on the similarity score in the model, and thus it is not surprising that many of the human participants ignored it.

On Problem 17, there is a great deal of disagreement, with 15% of participants choosing Answer 1 and 21% of participants choosing Answer 2, while 56% of participants choose Answer 4. Here again, the model predicts these results: While the model gives the highest score, 0.94, to Answer 4, it gives the scores 0.40 and 0.72, respectively, to Answers 1 and 2, making them the other two high-scoring answers. In this case, we suspect the large amount of disagreement among human participants was because this problem was particularly difficult, requiring an unintuitive mapping between pictures A and B. The model was only able to solve this problem by attempting alternate mapping modes (specifically the Alternative mode), while humans required 26.7 s, the most time of any problem, to solve it.

On Problem 19, 18% of participants chose Answer 2, rather than Answer 1. Here the explanation is straightforward: Problem 19 was ambiguous, with a different answer being preferred depending on whether one notices rotations or reflections first. While 82% of participants chose Answer 1, the correct answer if one sees the two ''A'' shapes as reflected, 18% of participants chose Answer 2, the correct answer if one sees the ''A'' shapes as rotated. Apparently most individuals, like our model, notice reflections before rotations, while a few notice the rotations first.

8. Related work

Evans' (1968) classic work on the ANALOGY system was the first to demonstrate that machines could do analogy. Since then, there have been other computational approaches to solving geometric analogy problems by Bohan and O'Donoghue (2000) and by Ragni, Schleipen, and Steffenhagen (2007). Both of these approaches differ from ANALOGY and our model in that they attempt to solve for D, the final picture in the analogy, directly, instead of picking from a list of possible answers. Unfortunately, the systems developed for these approaches appear to be fairly preliminary at this point. They are incapable of automatically identifying rotations and reflections between shapes, they require that some or all of the input representations be hand-coded, and they appear to actually require that the correspondences between shapes in pictures A and B be entered by hand. Furthermore, to our knowledge neither of these systems has been evaluated against human behavior. Bohand and O'Donoghue argue that their system, LUDI, improves on both Evans'

ANALOGY and structure mapping in that it solves geometric analogy problems that involve attributes of objects, such as colors and textures. This is incorrect; both ANALOGY and SME can handle attributes as easily as relations. The textured shape in Evans' Problem 13 is a prime example of an object attribute playing a key role in solving a problem.

Schwering, Krumnack, Kühnberger, and Gust (2007) present an alternate approach to solving geometric analogy in which only a single analogical mapping stage is used. Given hand-coded representations of each picture, the system begins by using Gestalt grouping principles to build up a representation of how pictures A and B relate to each other. This representation is then compared to picture C via a process of anti-unification, an approach to analogy in which two representations are compared and a common generalization of the two is identified. So far, their system has only been implemented over simple problems containing pairs of objects with no rotations or reflections, so it is unclear how the system would fare on more complex problems such as those described in this paper.

Our approach to geometric analogy is based on utilizing separate models for encoding and comparing stimuli (CogSketch and SME), each of which has been applied more generally across a number of other tasks. One alternative viewpoint is that encoding and comparison should be combined into a single, more domain-specific model. Tight interleaving of the construction of representations with analogical comparison is a hallmark of systems from Hofstader's collaborators, including Mitchell's (1993) Copycat program and French's (1995) TableTop program.[6] Tying the representation building and analogical reasoning processes together lets them interact more directly so that, for example, the representations can be updated dynamically, depending on the needs of the comparison. The disadvantage of this approach is that each system operates only in the domain for which it was designed, letter strings for Copycat and table settings for TableTop. The kinds of comparisons that can be made are hard-wired into the system. Similarly, Galatea (Davies & Goel, 2001) has a built-in specialized language of spatial entities and transformations that must be used in posing problems to it. By contrast, our work provides evidence that a general-purpose analogical matcher (SME) suffices for these tasks.

All the models presented thus far solve problems by using analogy to find patterns within a problem. In contrast, Eureka (Jones & Langley, 1995) solves problems by retrieving analogous situations in problems it has solved previously.

A number of general-purpose models of analogy have been constructed. Among these, the model most similar to SME is Keane and Bradshaw's (1988) IAM model, which is more serial than SME's parallel mapping process. Several connectionist models have been developed, including ACME (Holyoak & Thagard, 1989); LISA (Hummel & Holyoak, 1997), DRAMA (Eliasmith & Thagard, 2001), and CAB (Larkey & Love, 2003). Unfortunately, none of them appear to be able to handle the quantity of relationships that this task requires. Also, to the best of our knowledge, none of these systems have been applied to automatically constructed representations.

The problem of generating qualitative spatial descriptions was explored previously by Ferguson and Forbus (1999). They built GeoRep, a system which automatically constructed qualitative, symbolic descriptions of line drawings that could be used as the input for reasoning tasks such as recognition and identifying symmetry in a scene. Veselova and Davis (2004) showed how a system could learn a set of qualitative constraints describing a hand-drawn sketch that could then to be used to recognize other sketches of the same object. Their spatial representation scheme was not designed to be usable in any tasks other than recognition. Museros and Escrig (2004) constructed a system that generated qualitative descriptions of polygons that could be compared to identify rotations between the polygons. Their approach differs from our own in that they used a specialized comparison algorithm designed specifically for identifying rotations.

Our approach of decomposing a shape into edges based on discontinuities in the curvature takes some ideas from work on scale-space methods (Mokhtarian & Mackworth, 1986; Witkin, 1989), which analyze a curve at many different scales. In contrast, our edge segmentation algorithm operates at only three scales, which appears to be sufficient for most basic sketches. Furthermore, our approach decomposes a shape into discrete edges that can be related to each other qualitatively, which would be difficult with a scale-space representation.

9. Conclusion

Our results demonstrate that qualitative spatial representations and comparison via structure mapping can be used to perform geometric analogy. The model reported here selects the same answer as most human participants on the set of 20 geometric analogy problems from Evans, (1968). Furthermore, the model incorporates two preexisting components, CogSketch and SME, each of which has been used in a number of other tasks.

Further studies will be required before we can conclude with certainty that people solve spatial problems like geometric analogy via structure mapping. However, the simulation shows that two-stage structure mapping is sufficient for solving these types of problems. In addition, structure mapping over shape representations is sufficient for finding mappings between two shapes' edges in order to identify a rotation or reflection between them. Thus, the results support the claim that structure mapping plays a ubiquitous role in spatial problem solving. The high correlation between the number of first- and second-stage comparisons and human timing data on the problems provides additional support.

The comparison between the simulation and the human data also answers two of the open questions that we raised earlier in the paper. The first question was whether people identify a rotation or a reflection first when comparing two shapes. The model, in its current form, always prefers reflections. The human data suggest humans also prefer reflections, as the majority of participants chose the reflection-based answer on both of the ambiguous problems. However, all of the reflections in these problems were reflections over either the x- or y-axis. It remains to be seen whether people will show a similar preference for less regular reflections.

The second question was whether humans always align objects with the same shape when performing first-stage mappings. The results show that the Shape Identicality model, which required correspondences between entities with the same shape, predicted the human timing data better than the Normal model, which simply treated shape as another attribute. As noted above, we believe it still remains to be seen how strictly people apply this constraint. People may absolutely require that objects of the same shape align, under all circumstances, as is implemented in the Shape Identicality model. On the other hand, perhaps they simply strongly prefer to align objects with the same shape, but representations with enough structure supporting a shape mismatch could induce such an alignment. This might be implemented in our model by changing the model's representation of shape to give it a more central role. Additional experiments will be needed to tease apart these two possibilities.

One interesting distinction that can be made between the geometric analogy systems described in the Related Work section and our own model is that many of those systems attempt to solve directly for D, the answer which best completes the analogy, whereas our model (and Evans' original model) picks the best D from a list of possible answers. We view these as complementary approaches, and we suspect that people use both approaches in solving such problems. In some cases, the picture that completes the analogy may be particularly obvious, and people may think of it before even looking at the possible answers. However, in cases where the problem is more difficult, people may be more inclined to iterate over each possible answer and consider how well it completes the analogy. Note that several of the Evans problems were designed to be ambiguous; that is, there is more than one way to complete the analogy, and thus the only way to solve the problem is by considering which of the five listed answers best fits the analogy. We suspect that a hybrid system, which is capable of using either approach, depending on the difficulty of the problem, would most accurately model human behavior.

Our current model relies on a fixed set of mapping preferences for computing first-stage comparisons (e.g., Normal vs. Rotation-Preferred) and a fixed set of re-representation strategies for second-stage comparisons (e.g., abstracting from particular positional relations to PositionalRelation-Generic). While these suffice for this task, it seems likely to us that a broader range of options for mapping preferences and for re-representation will be necessary to handle the full range of geometric analogies that people can solve. We believe that modeling the process of learning these preferences and re-representation strategies from examples is a promising avenue for future work.

Notes

1. While SME can find more than three mappings, its default parameters limit it to three, as we have found this to be a reasonable number for use in cognitive simulations. In cases where one mapping is clearly more systematic than any others, only that one mapping will be produced.
2. To facilitate the large number of rotations that might exist between two shapes (e.g., four mappings between two squares), and thus the large number of mappings that might be returned, SME is allowed to return up to five mappings between the two shapes, instead of the normal maximum of three.

3. It is important to note that, unlike the normal positional relations (left-of, above), the positional-Generic relation is symmetric. This means that, when SME is finding a mapping between two positional-Generic relations, it will let their arguments match in either order. Thus, when positional-Generic is used to align a left-of relation with an above relation, the order of the arguments in the two relations does not matter.
4. The only difference in the parameters across these three uses of SME is that a greater maximum number of mappings is returned in the shape comparison, and shape identicality is enforced in the first-stage comparisons in Shape-Identicality mode.
5. Because of experimenter error, some participants were given the same random orderings. As many as five participants received one ordering, but on average only 1.5 participants received the same ordering. When we randomly selected one instance of each ordering, the number of participants dropped to 22, and the pattern of results remained the same.
6. O'Hara and Indurkhya's (1995) INA architecture used geometric analogy problems as a domain to further explore Hofstadter's claim that analogical mapping requires integrated description-building and mapping. Unfortunately, we cannot find papers describing either an implementation or an evaluation of INA.

Acknowledgments

This research was sponsored by the National Science Foundation as part of the Spatial Intelligence and Learning Center (Grant No. SBE-0541957), the Cognitive Science Program of the Office of Naval Research, and the Rapid Knowledge Formation program of the Defense Advanced Projects Research Agency.

References

Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, *94*, 115–147.

Biederman, I., & Gerhardstein, P. C. (1993). Recognizing depth-rotated objects: Evidence and conditions for 3D viewpoint invariance. *Journal of Experimental Psychology: Human Perception and Performance*, *19*, 1162–1182.

Bohan, A., & O'Donoghue, D. (2000). LUDI: A model for geometric analogies using attribute matching. *Proceedings of the eleventh artificial intelligence and cognitive science conference (AICS-2000)*. Galway, Ireland.

Cohn, A. (1996). Calculi for qualitative spatial reasoning. In J. Calmet, J. A. Campbell & J. Pfalzgraph (Eds.), *Artificial intelligence and symbolic mathematical computation* (pp. 124–143). London: Springer Verlag.

Davies, J., & Goel, A. (2001). Visual analogy in problem solving. *Proceedings of the 17th international joint conference on artificial intelligence (IJCAI-01)*. Seattle, WA.

Eliasmith, C., & Thagard, P. (2001). Integrating structure and meaning: A distributed model of analogical mapping. *Cognitive Science*, *25*(2), 245–286.

Evans, T. (1968). A program for the solution of geometric-analogy intelligence test questions. In M. Minsky (Ed.), *Semantic information processing* (pp. 271–353). Cambridge, MA: MIT Press.

Falkenhainer, B., Forbus, K., & Gentner, D. (1989). The structure–mapping engine: Algorithm and examples. *Artificial Intelligence*, *41*, 1–63.

Ferguson, R. W., & Forbus, K. (1999). GeoRep: A flexible tool for spatial representations of line drawings. *Proceedings of the thirteenth international workshop on qualitative reasoning (QR'99)*. Loch Awe, Scotland.

Forbus, K., Ferguson, R., & Gentner, D. (1994). Incremental structure-mapping. *Proceedings of the sixteenth annual conference of the cognitive science society*. Atlanta, GA.

Forbus, K., Ferguson, R., & Usher, J. (2001). Towards a computational model of sketching. *Proceedings of the 2001 international conference on intelligent user interfaces IUI'01*. Santa Fe, NM.

Forbus, K., & Oblinger, D. (1990). Making SME greedy and pragmatic. *Proceedings of the twelfth annual conference of the cognitive science society*. Cambridge, MA.

Forbus, K., & Usher, J. (2002). Sketching for knowledge capture: A progress report. *Proceedings of 2002 international conference on intelligent user interfaces IUI'02*. San Francisco, CA.

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*. Annecy, France.

Forbus, K., Usher, J., & Tomai, E. (2005). Analogical learning of visual/conceptual relationships in sketches. *Proceedings of twentieth national conference on artificial intelligence (AAA1-05)*. Pittsburgh, PA.

Forbus, K., Tomai, E., & Usher, J. (2003). Qualitative spatial reasoning for visual grouping in sketches. *Proceedings on the 17th international workshop on qualitative reasoning*, Brasilia, Brazil.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, *7*, 155–170.

Gentner, D. (1989). Mechanisms of analogical learning. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (pp. 199–241). London: Cambridge University Press.

Gentner, D., Brem, S., Ferguson, R. W., Wolff, P., Markman, A. B., & Forbus, K. D. (1997). Analogy and creativity in the works of Johannes Kepler. In T. B. Ward, S. M. Smith, & J. Vaid (Eds.), *Creative thought: An investigation of conceptual structures and processes* (pp. 403–459). Washington, DC: American Psychological Association.

Gentner, D., & Markman, A. B. (1997). Structure mapping in analogy and similarity. *Amercian Psychologist*, *52*, 45–56.

Gentner, D., Rattermann, M. J., & Forbus, K. (1993). The roles of similarity in transfer: Separating retrievability from inferential soundness. *Cognitive Psychology*, *25*, 524–575.

Holyoak, K. J., & Thagard, P. (1989). Analogical mapping by constraint Satisfaction. *Cognitive Science*, *13*(3), 295–355.

Hummel, J. E., & Holyoak, K. J. (1997). Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review*, *104*, 427–466.

Jones, R., & Langley, P. (1995). Retrieval and learning in analogical problem solving. *Proceedings of the seventeenth annual conference of the cognitive science society*. Pittsburgh, PA.

Keane, M. T., & Bradshaw, M. (1988). The incremental analogy machine: A computational model of analogy. In D. Sleeman (Ed.), *Third european working session on machine learning* (pp. 53–62). London: Pitman; San Mateo, CA: Morgan Kaufmann.

Klenk, M., Forbus, K., Tomai, E., Kim, H., & Kyckelhahn, B. (2005). Solving everyday physical reasoning problems by analogy using sketches. *Proceedings of the twentieth national conference on artificial intelligence*. Pittsburgh, PA.

Larkey, L., & Love, B. (2003). CAB: Connectionist analogy builder. *Cognitive Science*, *27*, 781–794.

Lovett, A., Lockwood, K., & Forbus, K. (2008). Modeling cross-cultural performance on the visual oddity task. *Proceedings of spatial cognition 2008*. Freiburg, Germany.

Lowe, D. G. (1989). Organization of smooth image curves at multiple scales. *International Journal of Computer Vision*, *3*, 119–130.

Markman, A. B., & Gentner, D. (1996). Commonalities and differences in similarity comparisons. *Memory & Cognition*, *24*(2), 235–249.

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. San Francisco: W.H. Freeman.

Mitchell, M. (1993). *Analogy = making as perception*. Cambridge, MA: MIT Press/Bradford.

Mokhtarian, F., & Mackworth, A. K. (1986). Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *8*(1), 34–43.

Museros, L., & Escrig, M. T. (2004). A qualitative theory for shape representations and matching. *Proceedings of the eighteenth international workshop on qualitative reasoning (QR'04)*. Evanston, IL.

O'Hara, S., & Indurkhya, B. (1995). Adaptation and redescription in the context of geometric proportional analogies. AAAI Fall Symposium on Adaptation of Knowledge for Reuse. Cambridge, MA.

Palmer, S. (1999). *Vision science: Photons to phenomenology*. Cambridge, MA: MIT Press.

Parsons, L. M. (1995). Inability to reason about an object's orientation using an axis and angle of rotation. *Journal of Experimental Psychology: Human Perception and Performance*, *21*, 1259–1277.

Ragni, M., Schleipen, S., & Steffenhagen, F. (2007). Solving proportional analogies: A computational model. *Workshop on analogies: Integrating multiple cognitive abilities, CogSci '07*, Nashville, TN.

Schwering, A., Krumnack, U., Kühnberger, K., & Gust, H. (2007). Using gestalt principles to compute analogies of geometric figures. *Proceedings of the twenty-nineth annual conference of the cognitive science society*. Nashville, TN.

Shepard, R. N., & Cooper, L. A. (1982). *Mental sketches and their transformations*. Cambridge: MIT Press.

Shepard, R. N., & Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science*, *171*, 701–703.

Tarr, M. J., Bülthoff, H. H., Zabinski, M., & Blanz, V. (1997). To what extent do unique parts influence recognition across viewpoint? *Psychological Science*, *8*(4), 282–289.

Veselova, O., & Davis, R. (2004). Perceptually based learning of shape descriptions. *Proceedings of the nineteenth national conference on artificial intelligence*. San Jose, CA.

Witkin, A. P. (1989). Scale-space filtering. *Proceedings of the eighth international joint conference on artificial intelligence (IJCAI-89)*. Detroit, MI.

Appendix

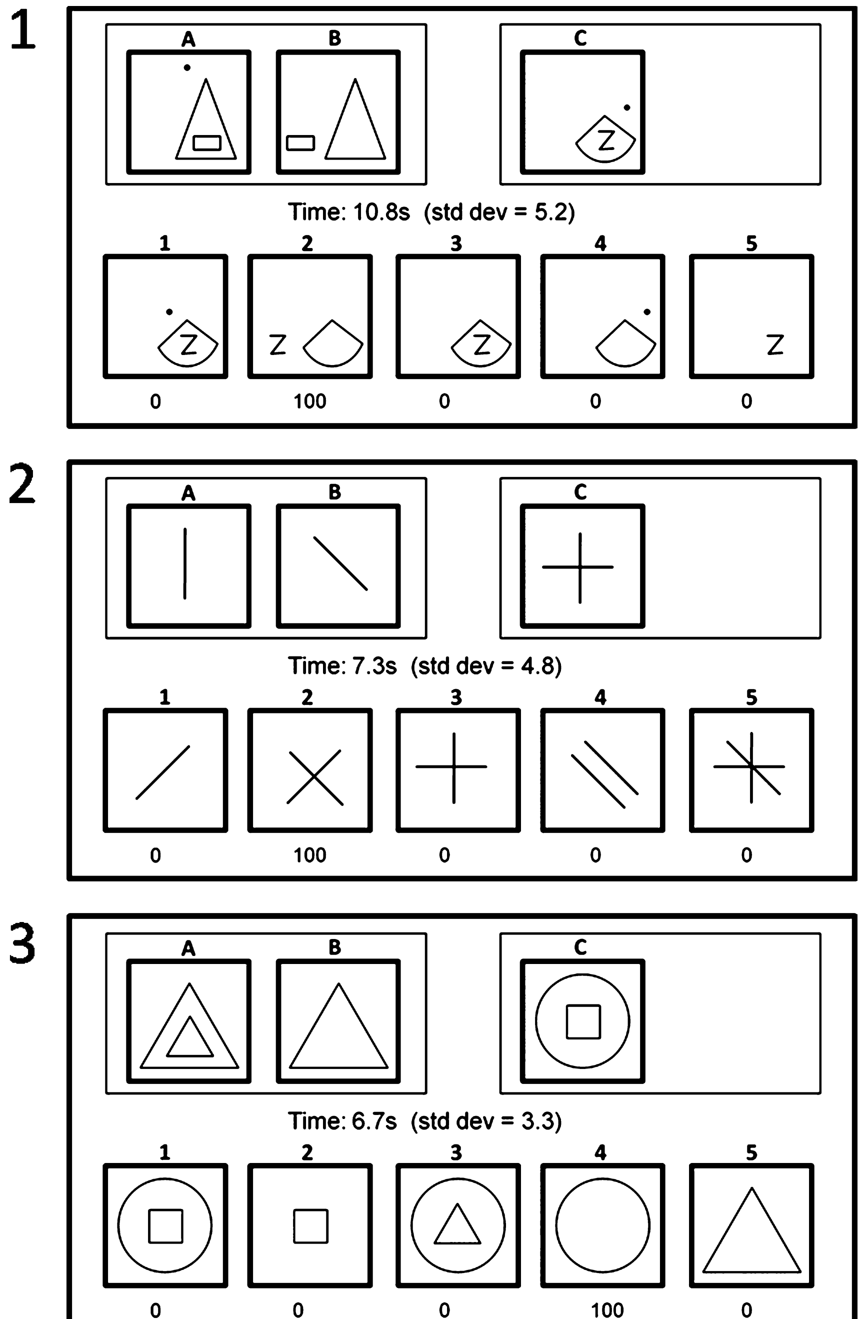


Fig. A1. Problems 1–3 (times are seconds required for human participants to pick an answer; values below answers are the percentage of participants who picked each answer).

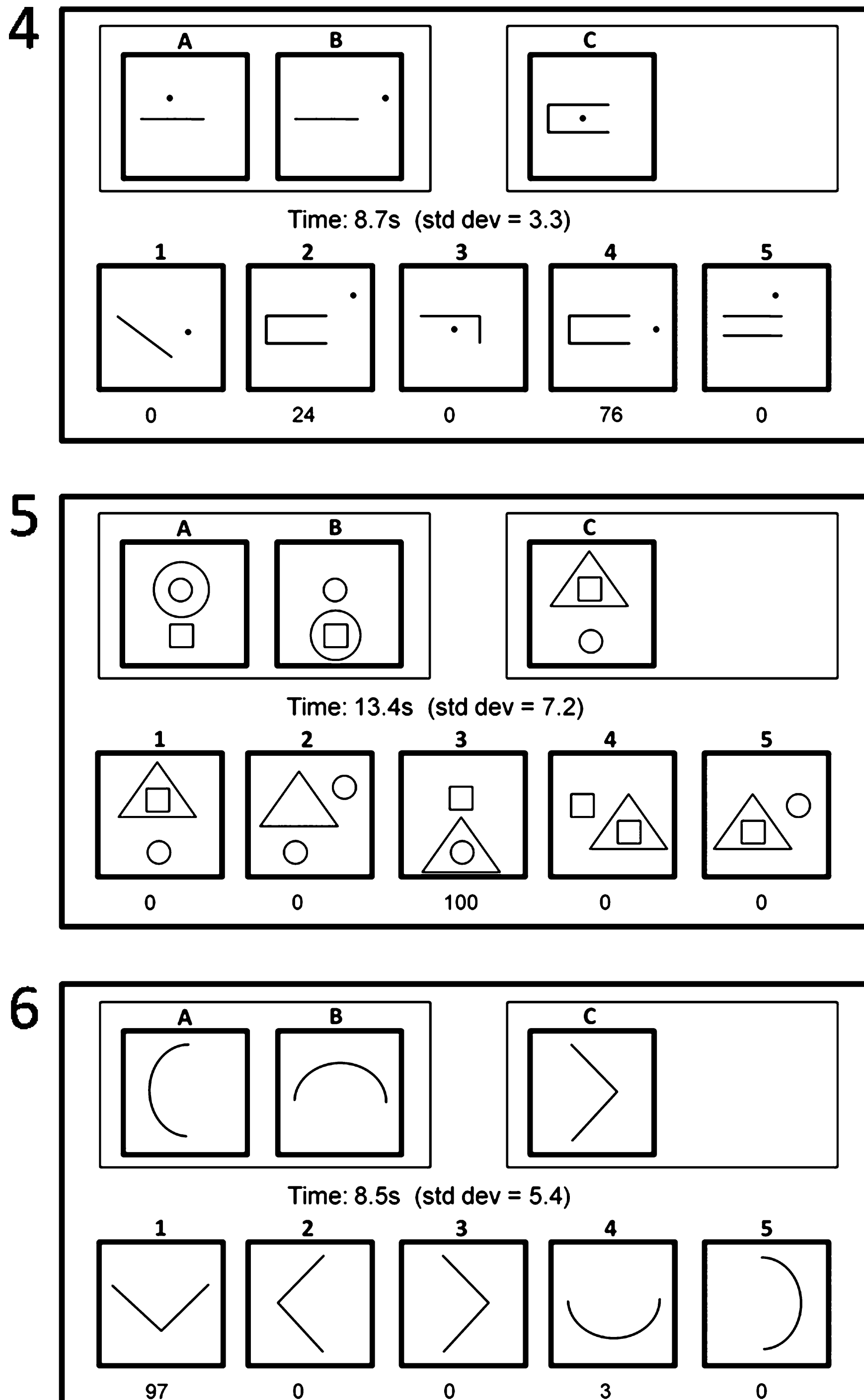
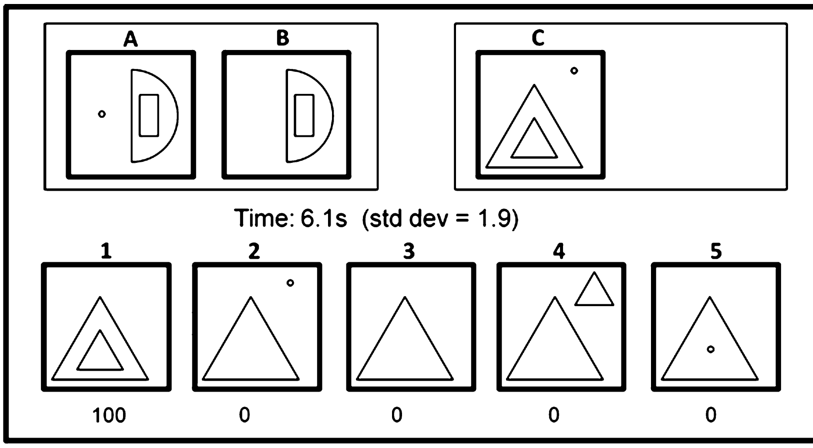
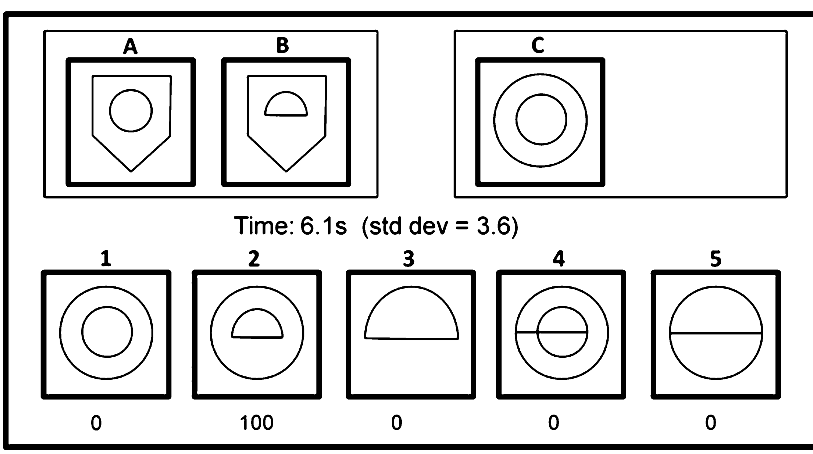


Fig. A2. Problems 4–6.

7



8



9

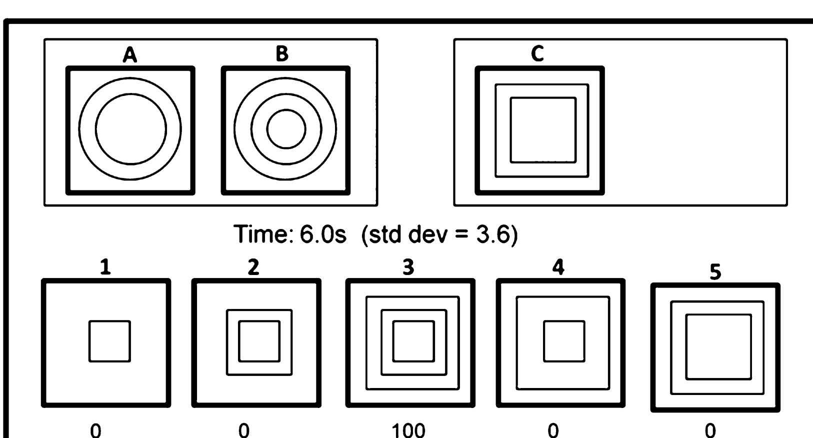
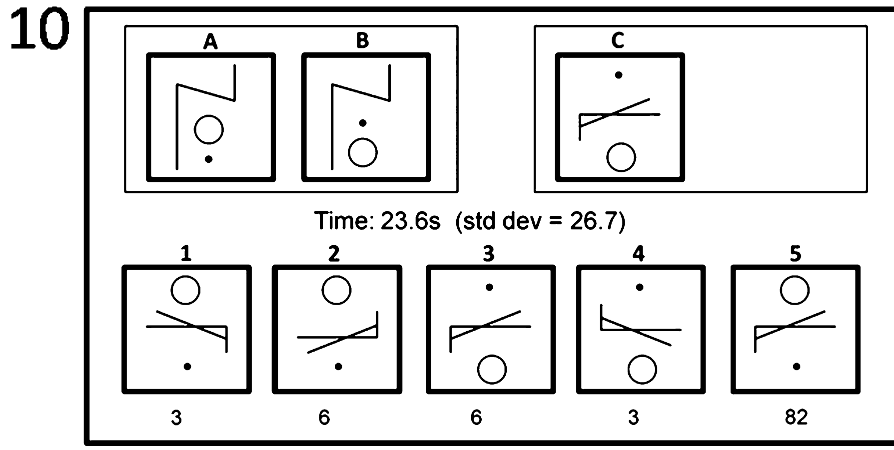
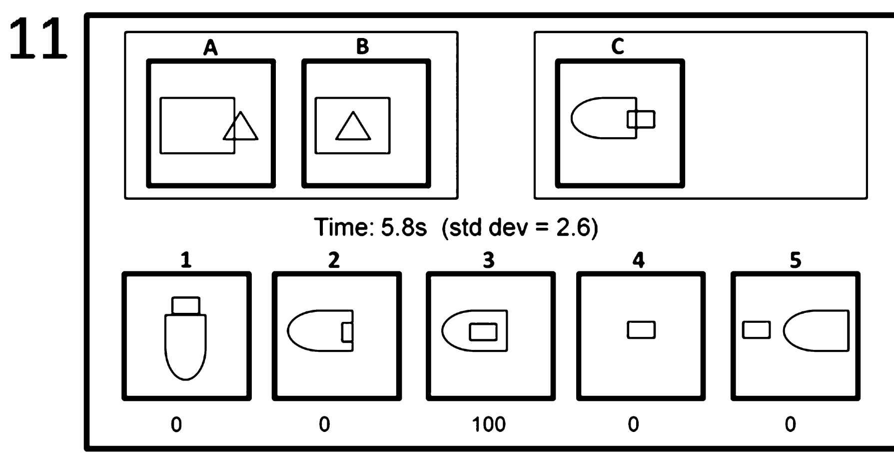


Fig. A3. Problems 7–9.





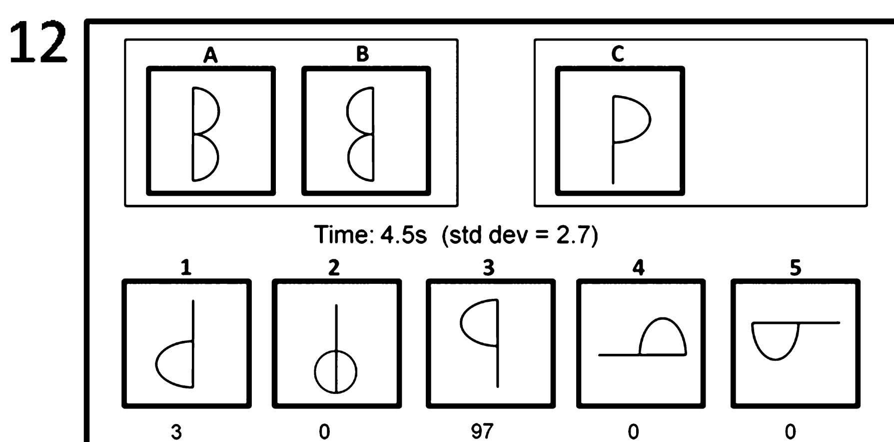


Fig. A4. Problems 10–12.

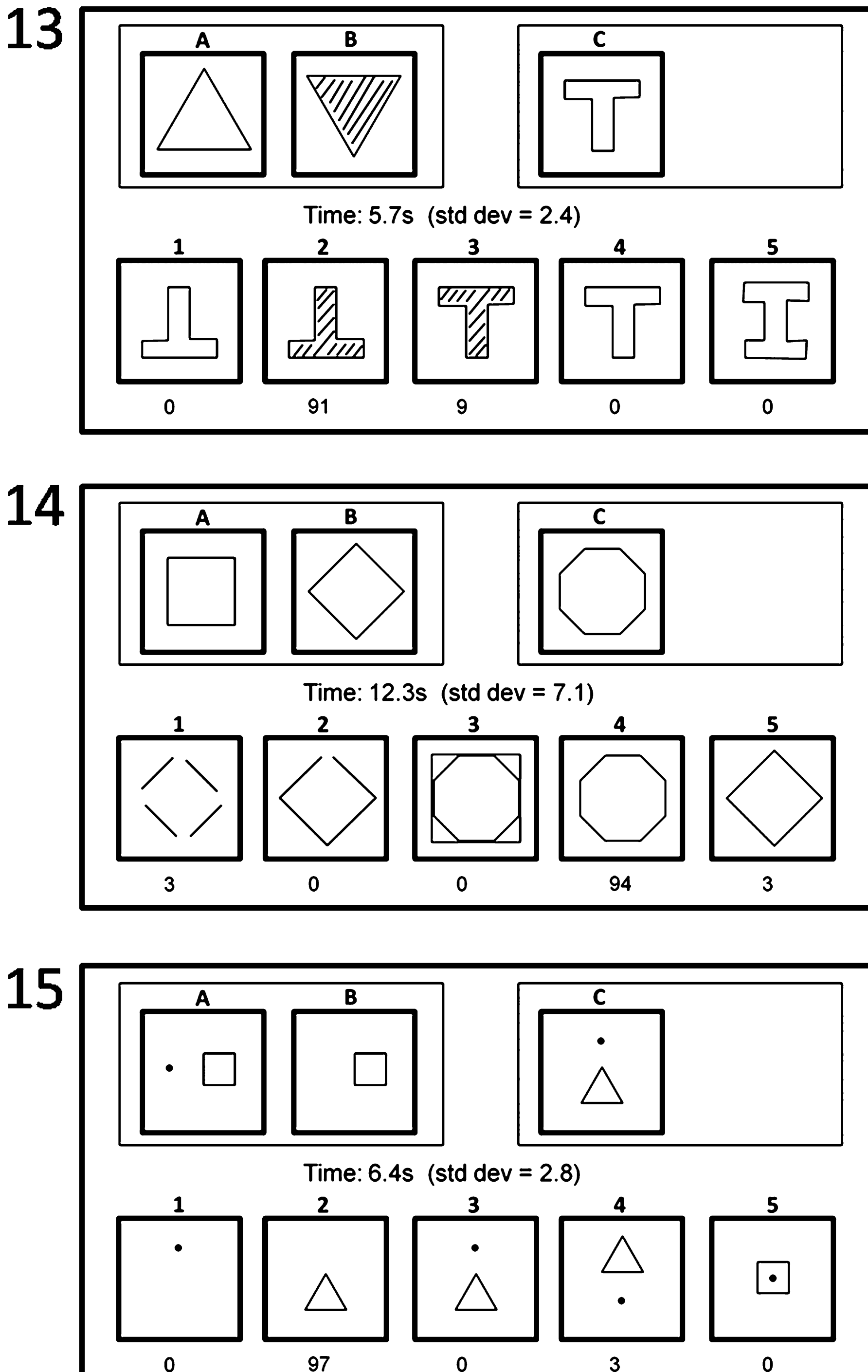
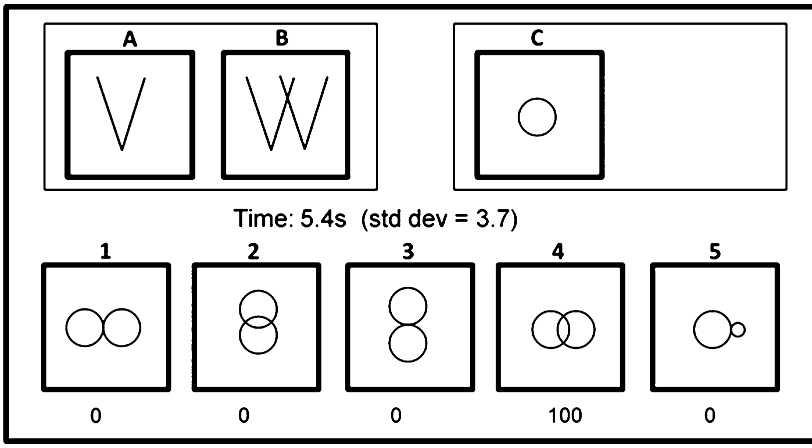
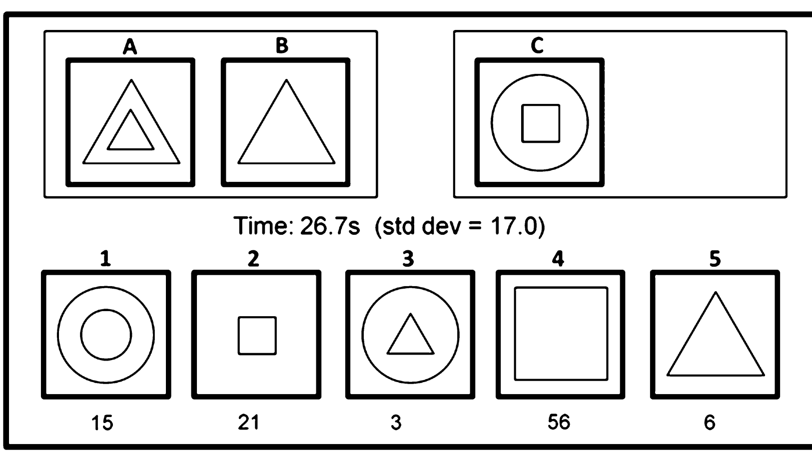


Fig. A5. Problems 13–15.

16



17



18

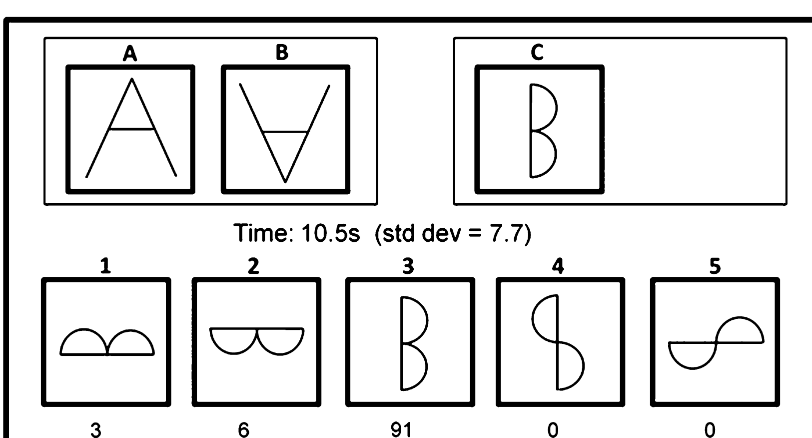
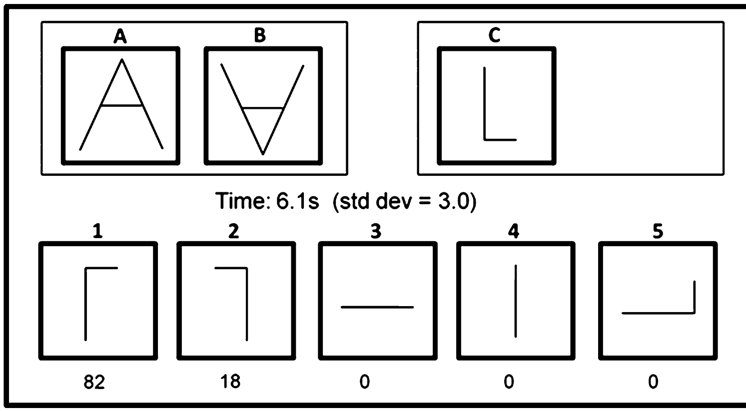


Fig. A6. Problems 16–18.

19



20

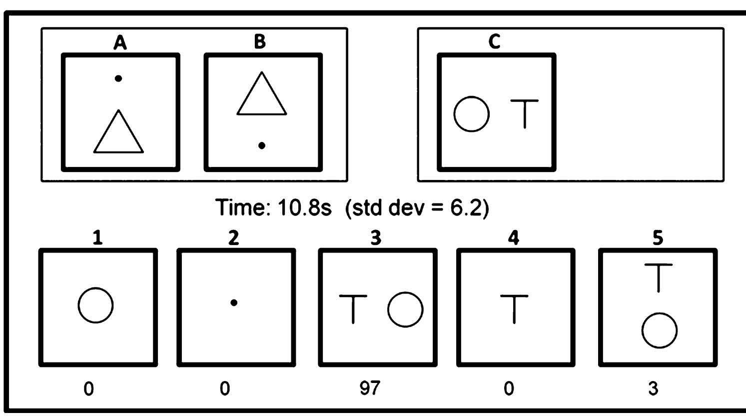


Fig. A7. Problems 19–20.