Synthesis of Solutions for Shaded Area Geometry Problems

Chris Alvin,[1] Sumit Gulwani,[2] Rupak Majumdar,[3] Supratik Mukhopadhyay[4]
[1] Bradley University calvin@bradley.edu
[2] Microsoft Research, Redmond sumitg@microsoft.com
[3] MPI-SWS rupak@mpi-sws.org
[4] Louisiana State University supratik@csc.lsu.edu

Abstract

We motivate and address the task of automatically solving shaded area geometry problems. Our approach consists of identifying atomic regions in a coordinate-based geometry figure, building an *analysis hypergraph* representing all facts that can be derived of the figure (using saturation based reasoning), and then finding a path in the hypergraph from the given facts to the goal. On a corpus of 102 problems taken from popular high-school geometry textbooks, our tool *GeoShader* successfully solved and characterized all problems in an average time of 13.4 seconds.

1 Introduction

We describe *GeoShader*, a tool that can solve *shaded area* geometry problems. A shaded area problem is composed of a geometric figure, a set of given facts about that figure, and a shaded region in the figure whose area is to be found. See Figure 1 for a sample shaded area problem.

A solution to a typical high school geometry problem requires a student to use deductive logic while reasoning about the visual elements in a given figure. Shaded area problems go a step further by requiring recall of formulae for different shapes as well as exercising the associated quantitative skills necessary to compute the area of the desired region.

While a typical shaded area problem is quite demanding of a student to exercise their skills, it has a clear quantitative answer. This makes shaded area problems ideal for multiple choice problems compared to geometry construction or proof problems that have many possible solutions and require expert knowledge to assess a solution. It becomes clear why shaded area problems are often encountered on standardized high school Mathematics examinations (e.g., ACT, SAT, State Comprehensive Assessments (Massachusetts DOE 2014; NY State Education Dept. 2014), etc.) and even on some graduate level ones (e.g., GRE quantitative).

One can represent the solution process for a shaded area problem as a directed acyclic graph (DAG), where each node represents intermediate facts that are true of the figure (and are derivable from the predecessor nodes), and are useful for computing the goal area. A solution to a shaded area problem thus has quantifiable features corresponding to properties of its solution DAG. For example, a solution (and its corresponding problem) has depth and width. *GeoShader* computes the structural features of a solution as well as other descriptive features of a solution, including a level of difficulty corresponding to the number of deductions, geometric facts, and facts related to area. Each of these features gives a teacher the ability to effectively identify or compare problems (with associated solution) when constructing homework or exams.

GeoShader solves a shaded area problem by first dissecting the given figure into its closed, constituent areas using a planar graph-based representation. It then arranges the shapes in the figure into a hierarchy followed by a fixed-point technique to acquire the area of regions in the figure. The area of a region is thus a linear combination of areas of other regions. *GeoShader* represents all possible algebraic decompositions as a hypergraph in which the solution to a problem can be obtained by traversing this hypergraph.

We evaluated the effectiveness of our solution techniques on 102 figures taken from popular geometry textbooks and exams. *GeoShader* solved each problem in an average of 13.4 seconds.

We make the following contributions:

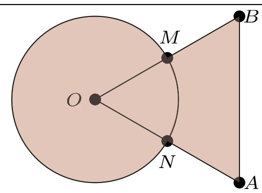
- We formalize the notion of a shaded area geometry problem (along with some useful features associated with it) and its solution (§2).
- We describe an algorithm to efficiently solve shaded area problems from existing figures (§3).
- We describe experimental results illustrating the efficacy of our problem solving algorithm (§4).

2 Formalization

2.1 Problem Definition

Facts. Solving a shaded area problem requires manipulating two kinds of facts—geometric facts and facts about areas. A *geometric fact* for a figure Fig is a logical proposition about the figure, such as "OAB is an equilateral triangle" or "$OM = 7$." (We omit units for readability.) We refer to the geometric facts of Fig as $\mathcal{E}(\mathsf{Fig})$. An *area fact* for a figure Fig is of the form "$\mathsf{Area}(g) = c$" where g is a closed region

Find the area of the shaded region with circle radius 7cm and equilateral triangle OAB of side length 12cm shown at right. (CBSE, India 2012)



The goal region g is the entire figure with explicit facts $\{\mathsf{EqTri}(O, A, B), OM = 7, OA = 12\}$. The solution summing the areas of $\mathsf{MajSector}(M, O, N)$ and $\mathsf{EqTri}(O, A, B)$ is depicted as a directed hypergraph.

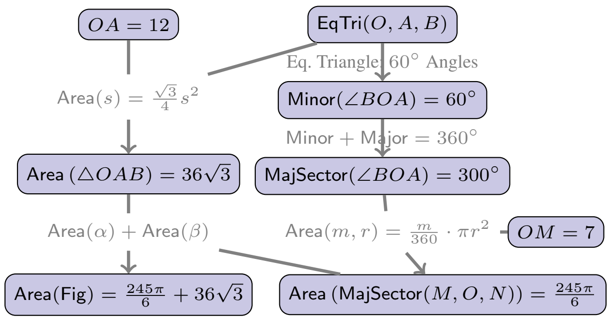


Figure 1: Example Shaded Area Problem and Solution

in the figure and c is a numeric constant. We refer to the area facts of Fig as $\mathcal{A}(\mathsf{Fig})$.

Formal Definition. A *shaded area geometry problem* is a triple consisting of (i) a geometric figure, (ii) a set of assumptions about that figure, and (iii) a goal area to compute. A (geometric) *figure* is a fixed arrangement of geometric objects (e.g., circles, polygons, etc.) in a specific orientation in the Euclidean plane. The *set of assumptions* consists of geometric facts that may be described as measurement facts (e.g. measure of an angle is 45^o, length of side is 7cm) and facts relating objects in the figure (e.g. two angles are congruent, points are collinear, etc.). The *goal* is defined by a region in the figure for which we wish to compute the area.

The objective is to compute the area of the shaded portion using geometric reasoning (i.e., logical reasoning using the given facts and the axioms of geometry), area computations of shapes (e.g., computing the area of a circle with known radius), and algebraic manipulations (e.g., expressing the area of a region as the sum or difference of two other regions).

Definition 1 (Shaded Area Problem)**.** *A* shaded area problem *P is a triple $P = \langle \mathsf{Fig}, A, g \rangle$ where Fig is a geometric figure, $A \subset \mathcal{E}(\mathsf{Fig})$ is a set of assumptions, and g is an area fact for a region in Fig.*

2.2 Solution to a Shaded Area Problem

A solution to a shaded area problem is based on manipulating facts about a figure and deductions connecting those facts.

Deductions. Computing a shaded area requires three kinds of reasoning: (a) deductive reasoning about geometric facts (e.g., deducing the length of a radius of a circle given the length of another radius), (b) computing the area of a shape once its parameters are known, and (c) algebraic composition of areas using sums and differences of areas.

Deductions about geometric facts follow first-order geometric reasoning using Hilbert's axioms of geometry and are annotated with the deduction rule used to derive the conclusion. For example, in Figure 1 we derive the measure of angle $\angle BOA$ is $60°$ from the fact that $\triangle OAB$ is equilateral; we label the edge with the appropriate axiom.

Deductions about area facts are of two forms. The first computes the area of a region given a set of geometric facts. In Figure 1 the circle with center O and (known fact) radius $OM = 7$, we deduce from the facts $\mathsf{Circle}(O, OM)$ and $OM = 7$ a fact labeled $\mathsf{Area}(Circle(O, OM)) = 49\pi$.

The second form computes the area of a region by algebraic manipulation of other areas. For example, we may deduce from the area facts $\mathsf{Area}(\mathsf{MinSector}(M, O, N)) = \frac{1 \cdot 49\pi}{6}$ and $\mathsf{Area}(\mathsf{MajSector}(M, O, N)) = \frac{5 \cdot 49\pi}{6}$ the fact that $\mathsf{Area}(\mathsf{Circle}(O, OM) = 49\pi$ because the sum of the areas of the former facts deduces the area of the latter. Again, we label this deduction with an algebraic expression stating the conclusion is derived by adding the premises.

Solution. The *solution* of a shaded area problem consists of computing the area of the goal region g, if possible, using the set of implicit facts (those facts that are independent of scaling the figure, e.g., OAB is triangle in Figure 1) and facts (lengths of line segments or radii) and deductions of the types described above.

A solution to a shaded area problem takes the form of a DAG with one leaf node (which represents the goal) and multiple root nodes (which represent the facts stated in the problem). Furthermore, each non-root node is labeled with a deduction rule that denotes the rule used to derive that node from all of its predecessor nodes.

Definition 2 (Solution to a Shaded Area Problem)**.** *For a shaded area problem $P = \langle \mathsf{Fig}, A, g \rangle$, the* solution *to P is a directed acyclic graph (N, D) where $N \subset \mathcal{E}(\mathsf{Fig})$ are the facts (nodes) required to deduce g from the set of assumptions A and D is the set of edges corresponding to logical deduction among facts.*

Example. Figure 1 is an example of a shaded area problem and its solution. It is given that triangle OAB is equilateral, and further, the lengths of the radius of the circle and the side of the triangle. To compute the area of the shaded region, one can perform algebraic decomposition to express the shaded region as the sum of the areas of the major sector and the triangle. Using the length of the side and the elementary formula for the area of an equilateral triangle, we compute the area of the triangle as $36\sqrt{3}$. To compute the area of the major sector MON, we need two deduction steps: the measure of minor angle $\angle MON$ is $60°$ and thus the measure of the major $\angle MON$ is $300°$. Since the radius of the circle is known, we compute the area of the major sector as $\frac{300}{360} \cdot \pi r^2 = \frac{245}{6}\pi$. Now, the area of the shaded region is obtained using algebraic manipulation.

Each of these deductions steps are clear in the hypergraph representation of the solution. Structurally, we see a path from the source nodes (corresponding to the problem as-

The figure at right contains four atomic regions labeled (1)-(4).

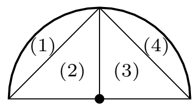


Figure 2: Labeled Atomic Regions for a Figure

sumptions) to the single goal node; this path constitutes a problem solution.

3 Solving Shaded Area Problems

We now describe *GeoShader*, our tool for solving shaded area problems. Its input consists of a geometric figure drawn to scale in the Euclidean plane. We compute an internal, directed hypergraph representation of the figure using a coordinate-based analysis. Then, we traverse the hypergraph to identify a solution to a given shaded area problem.

3.1 Preliminaries

Given a geometric figure Fig and a set of geometric axioms Axm, (Alvin et al. 2014) describe a logical, *saturated hypergraph* in which nodes correspond to the geometric facts of Fig and whose edges are of the form (S, t, A) where t is a fact deduced from a set of facts S and annotated with geometric axiom $A \in \mathsf{Axm}$. Our solving technique extends this notion of a *saturated hypergraph* to additionally track facts about areas in an analysis hypergraph. We also extend the set of geometric axioms to include *shape axioms*; that is, a common geometric formula for calculating the area of a shape (e.g. the area of a rectangle is the product of length and breadth).

Definition 3 (Analysis Hypergraph)**.** *An* analysis hypergraph $H(\mathsf{Fig})$ *for a figure* Fig *is a directed hypergraph whose nodes consist of all geometric facts* $\mathcal{E}(\mathsf{Fig})$ *and area facts* $\mathcal{A}(\mathsf{Fig})$ *in* Fig *and whose edges are of the form* (S, t, A)*, where* $S \subseteq \mathcal{E}(\mathsf{Fig}) \cup \mathcal{A}(\mathsf{Fig})$ *is a set of geometric facts or area facts called the* source nodes*,* $t \in \mathcal{E}(\mathsf{Fig}) \cup \mathcal{A}(\mathsf{Fig})$ *a geometry or area fact called the* target node*, and* A *a geometric axiom such that* A *deduces* t *from* S*.*

For a figure, we are interested in the set of smallest, closed components of a figure we call *atomic regions*; Figure 2 consists of four such components (numbered (1)-(4)). Consistent with the definition of a Jordan curve (Jordan 1893), we note that atomic regions may be convex ($\triangle OAB$ in Figure 1) or non-convex (major sector defined by MNO in Figure 1). A shaded area problem may 'shade' several atomic regions in a figure such as the entire figure in Figure 1; a *region* is a non-empty set of atomic regions.

Definition 4 (Shape, Region, Atomic Region)**.** *A* shape *is a geometric object (e.g., square, circle, triangle) for which we can directly calculate the area using some pre-defined geometric formula, provided appropriate parameters are known. An* atomic region *is a Jordan curve that cannot be further decomposed into two closed regions by an existing line or arc passing through it. A* region *is a non-empty set of atomic regions.*

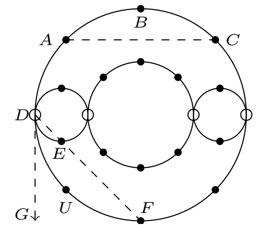


Figure 3: Annotated planar graph of a figure.

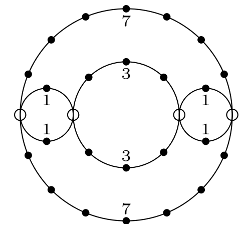


Figure 4: Automated planar graph of a figure. Values indicate the number of points added along the arc.

We note that a region may or may not be a shape and thus may or may not have a directly computable area. We also observe that the set of all regions for a figure Fig is the powerset of atomic regions (save the emptyset). It suffices to compute the atomic regions for a given figure to compute the regions and thus the corresponding set of area facts.

3.2 Atomic Region Identification

We now consider how to identify the set of atomic regions in a given figure. The analysis hypergraph is based on geometric and area facts. We compute geometric facts by analyzing the coordinate-based figure. The goal of a given shaded area problem is based on the area of some, potentially disconnected, set of closed geometric objects. In order to compute area facts (nodes in the analysis hypergraph), we must first identify the atomic regions in the given figure. In this subsection we describe how we compute these atomic regions for a particular input figure.

The atomic region identification algorithm takes a geometry figure Fig and computes all atomic regions by constructing a planar graph whose vertices are intersection points[1] in a figure Fig and whose edges are segments which arise from non-crossing lines and arcs of the figure. Atomic regions are then the facets of this planar graph—the smallest, closed regions in the planar graph corresponding to atomic regions in the geometry figure.

Given a planar graph, our algorithm processes all points in lexicographic order. For each point, it greedily chooses an edge in the counterclockwise order to extract facets (Edelsbrunner 1987), removes the first edge from the graph, and repeats the facet identification process. Unfortunately, the planar graph obtained by only considering the explicit points and lines or arcs in a figure do not uniquely determine the figure; ambiguities may arise when a figure has arcs.

Resolving Arc Ambiguity. First, we must be able to distinguish between segments and arcs. In Figure 3 there is no distinction between chord $\overline{AC}$ and arc 6.0pt$\widehat{AC}$ in a corresponding planar graph. The addition of point B successfully disambiguates $\overline{AC}$ from 6.0pt$\widehat{AC}$.

[1]Intersection points are those points attributed to one shape intersecting another shape in a given figure.

A second issue arises when we traverse a planar graph to identify the facets. Consider the planar graph corresponding to Fig in Figure 3; ambiguity arises in this case when we begin traversal from point D. The problem is attributed to the fact that points D, E, and F are collinear and thus create the same counterclockwise angle with respect to the reference vector $\overrightarrow{DG}$ (i.e., $\angle FDG \cong \angle EDG$). Starting at D, the next point in a counterclockwise traversal should be F. This is due to the fact a greedy traversal chooses the edge with the smallest counterclockwise angle with respect to $\overrightarrow{DG}$; in this case, $\angle UDG < \angle EDG$. However, without the inclusion of the point U along 6.0pt$\widehat{DF}$ in Figure 3, the choice of the next edge from D is ambiguous: we cannot distinguish the edge D to E from the edge from D to F.

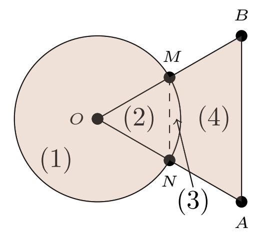
Resolving both ambiguities requires addition of points along each arc. Figure 3 demonstrates one successful addition of points for the planar graph to uniquely determine the figure. Our approach to resolving these ambiguities is based on the number and size of circles in the figure. Sorting the circles by radius length from least to greatest, we add an exponentially increasing number of points at constant intervals along all arcs: 1, 3, 7, etc. as shown in Figure 4; each arc in the two smallest circles have one additional point added, 3 points for each arc in the middle circle, etc. The planar graph corresponding to our automatic technique is shown in Figure 4 where ‘open’ points are intersections points among the shapes and dark points are the additional points added to resolve ambiguities.

3.3 Constructing the Analysis Hypergraph

In this section we describe two techniques to deduce area facts. First, we identify shapes in the given figure in order to deduce an area fact from geometric facts and shape axioms. Second, we must note that deducing an area fact from two area facts by means of addition or subtraction of the respective areas is a simple process, but is computationally expensive. This is due to the fact that the number of facets of a planar graph is linear in the size of the graph while the number of regions, corresponding to sets of facets, is exponential. We therefore do not construct the entire analysis hypergraph for a given figure, but limit construction of nodes to the set of assumptions in the problem. That is, we use the following algorithm as a heuristic to avoid area facts that are not computable with the problem parameters. We deduce an area fact from sets of area facts using an algorithm composed of two parts. (1) Organize the shapes into a hierarchy, computing areas of regions traversing down the hierarchy. (2) Use a fixed-point approach to compute areas of regions that are unions or differences of two regions by respectively adding or subtracting known areas.

Deducing Area Facts from Geometry Facts (Regions as Polygons). We cannot directly compute the area of all regions since a region may not be a shape or a region may contain atomic regions that are disconnected. However, we must identify all shapes so that we can compute the area of the corresponding region, if the parameters are known. We can mathematically compute the area of arbitrary polygons using coordinate geometry techniques (Edelsbrunner 1987);

Consider figure Fig from Figure 1 annotated at right. For *atomic region identification*, we construct chord $\overline{MN}$ resulting in four atomic regions labeled (1)-(4) (thus $2^4 - 1 = 15$ regions).



The corresponding *shape hierarchy* for Fig consists of a circle, two sectors, a trapezoid, and two triangles.

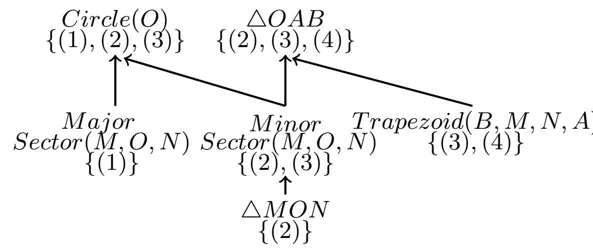


Figure 5: Solving the shaded area problem from Figure 1.

however, solving shaded area problem solutions are based on classic geometric formulas for standard polygons such as triangles (half base multiplied by height) or rectangles (base multiplied by height). We present an algorithm for identifying all polygons (convex and non-convex) in a given figure.

From our coordinate-based analysis of a figure we have the corresponding set of all segments. We first identify candidate segments which may be combined into a polygon by eliminating invalid combinations of segments that do not share a vertex or are collinear. Second, we exhaustively construct the set of all triangles in the figure from the set of valid, closed combinations of three segments. Last, we inductively construct polygons of increasing numbers of sides by considering valid sets of segments that do not contain any previously established polygon.

Given the set of polygons, circles, and sectors, we can match the shape with the corresponding region. That is, we construct a hyperedge from the parameters (source nodes) used to compute the area fact (target node) labeling with the geometric area formula used in the deduction. We note in Figure 5 that $\triangle OAB$ corresponds to region $\{(2), (3), (4)\}$ and in Figure 1 we use the geometric facts $OA = 12$ and $\mathsf{EqTri}(O, A, B)$ to deduce $\mathsf{Area}(\triangle OAB) = 36\sqrt{3}$ by way of the area formula for an equilateral triangle.

Deducing Area Facts from Area Facts I (Shape Hierarchy). Instead of exhaustively exploring all possible relationships among subsets of atomic regions, we use a hierarchy of shapes as a heuristic. We organize sets of atomic regions as a directed acyclic graph called the *shape hierarchy*. The roots of the shape hierarchy are shapes we will refer to as *root shapes*. We construct the shape hierarchy by noting that the children of a node are shapes that are fully contained in their parent; we link only direct containment. In Figure 5, $\mathsf{MinSector}(M, O, N)$ is directly contained within both $\mathsf{Circle}(O, OM)$ and $\mathsf{Triangle}(O, A, B)$ so there exists directed edges in the associated shape hierarchy from $\mathsf{MinSector}(M, O, N)$ to both shapes.

Given a shape hierarchy, for a figure Fig, our first step is a traversal of the shape hierarchy to compute the areas of the shapes, if possible, and labeling the shape hierarchy accordingly. These computations require additional geometric reasoning (e.g., to find out the length of a line segment from given facts about the figure). The second step mimics how a student may approach handling area calculations by taking a series of differences. For each shape in the hierarchy, we individually subtract each of its descendants to acquire a region and its associated area, if possible. For example, in Figure 5, we can compute the area of region $\{(1), (3)\}$ by taking the difference between $\mathsf{Circle}(O, OM)$ and $\mathsf{Triangle}(M, O, N)$: $\mathsf{Area}(\{(1), (3)\}) = \mathsf{Area}(\mathsf{Circle}(O, OM)) - \mathsf{Area}(\mathsf{Triangle}(M, O, N))$. Similarly, we may compute the area of region $\{(3), (4)\}$ which defines the $\mathsf{Trapezoid}(B, M, N, A)$ as $\mathsf{Area}(\{(3), (4)\}) = \mathsf{Area}(\mathsf{Triangle}(O, A, B)) - \mathsf{Area}(\mathsf{Triangle}(M, O, N))$.

Deducing Area Facts from Area Facts II (Fixed-Point Algebraic Combinations). Our last step in constructing area facts computes areas of additional regions using a fixed-point computation. Given the areas of regions acquired through the shape hierarchy analysis, we exhaustively combine all regions and their associated areas. That is, in each iteration, we take two regions whose areas have been computed. If the regions are disjoint, we take the sum to identify the area of the union of the two regions. In the case where one region is completely contained within the other region, we compute the area of the difference of the two regions.

Finding the area of a goal region in some shaded area problems does not require this step; however, in the case of Figure 5 solving the problem is impossible without this fixed-point combining process. In Figure 5 we know $\mathsf{Area}(\mathsf{MajSector}(M, O, N)) = \frac{245}{6}\pi$ and $\mathsf{Area}(\triangle OAB) = 36\sqrt{3}$ with respective regions $\{(1)\}$ and $\{(2), (3), (4)\}$. Taking the union of the two regions results in the solution to the problem $\mathsf{Area}(\mathsf{Fig}) = \mathsf{Area}(\{(1), (2), (3), (4)\}) = \frac{245}{6}\pi + 36\sqrt{3}$. *GeoShader* solves the problem in Figure 1 via this algebraic combining process.

3.4 Finding a Path in the Hypergraph

As noted in §2.2, a solution to a shaded area problem is a path in the corresponding analysis hypergraph. Our goal is to identify such a path for some shaded area problem $P = \langle \mathsf{Fig}, A, g \rangle$. Identifying a solution to problem P consists of two distinct steps. The first step takes Fig and uses the process described in §3.3 to construct the corresponding analysis hypergraph, $H(\mathsf{Fig})$. The second step to solve problem P is to identify a path from the nodes corresponding to A and the goal node corresponding to g in $H(\mathsf{Fig})$. The resultant solution is the DAG described in §2.2.

4 Experimental Results

Evaluation Criteria. We first describe our benchmark set of problems and characteristics of the corresponding figure. Second, we evaluate our solution technique with respect to

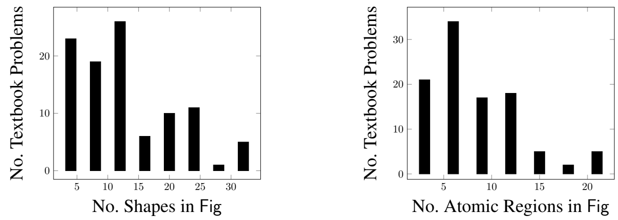


Figure 6: Characteristics of textbook problems.

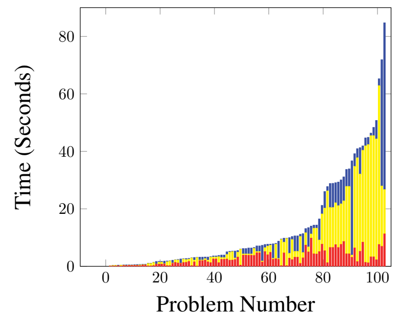


Figure 7: (Sorted) Times for Solving Shaded Area Problems: Atomic Region Identification (red, gray in grayscale), Deduction Engine (yellow, light in grayscale), and Computing the Solution and its Features (blue, dark in grayscale).

time required to construct the corresponding hypergraph and identify the solution path. Last, we correlate structural characteristics of a solution with respect to the time taken to generate that solution. We execute our solution generation algorithm on a laptop with Intel Core i5-2520M CPU at 2.5GHz with 8 GB RAM on 64-bit Windows 7 operating system.

Benchmark Problem Set. We ran our solution generation algorithm on a set of 102 figures taken from standard mathematics textbooks and workbooks from the United States (Jurgensen, Brown, and Jurgensen 1988; Holt, Rinehart, and Winston 2007; Larson et al. 2007; C. Boyd 2006; Chew 2008) as well as released exams from the Indian Class X exam (CBSE, India 2012). Each of the 102 problems we solved successfully. We used a uniform set of geometric area formulas and geometric axioms for all of our experiments: tangents, circles, quadrilaterals, congruent triangles, etc.

In the set of 102 figures from textbook problems we observe a figure in a shaded area problem has mean (and standard deviation) 11.5 (7.8) shapes and 7.3 (4.6) atomic regions. We note that for many shaded area problems, there are often more shapes than atomic regions; as an example, consider the six shapes and four atomic regions in Figure 5. The number of shapes and atomic regions per problem result in right-skewed distributions as shown in Figure 6.

Problem Solving vs. Time. Solving a shaded area problem (§3) requires computing the atomic regions in the figure (§3.2), construct the logical hypergraph, constructing the analysis hypergraph (§3.3) and path identification of the solution (§3.4); Figure 7 shows the time required for each of the three phases. We note a mean (and standard devia-

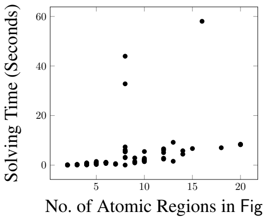


Figure 8: No. Atomic Regions vs. Solving Time.

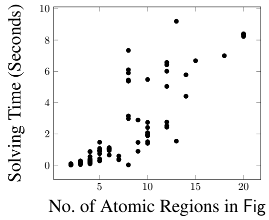


Figure 9: Three Outliers Removed from Figure 8.

tion) of 2.79 (2.53) seconds for atomic region identification, 7.29 (12.10) seconds for deduction engine construction, 3.33 (7.91) seconds for area fact deduction and computing the solution, and overall time 13.4 (17.24) seconds.

Constructing the nodes and edges in an analysis hypergraph, which avoids eager consideration of the exponential number of regions, is thus well-motivated since the number of atomic regions can often be large. The raw data in Figure 8 leads to an inconclusive relationship due to three outliers we identified as having to construct the complete analysis hypergraph. Instead, consider the linear correlation ($r^2 = 0.721$) or monomial-power correlation ($r^2 = 0.782$) in Figure 9. In either case, solving a shaded area problem in polynomial order of atomic regions is noteworthy.

Solution Characteristics. The solution to a shaded area geometry problem is a DAG (§2.2) and therefore has several quantitative features. For example, the *depth* of a solution is the longest path from the assumptions to the area in the solution, *width* is the maximal number of nodes in a level, and the number of *deduction steps* corresponding directly to the number of hyperedges in the solution. With our solutions to the 102 shaded area problems, we see a mean (std. dev.) for depth 7.0 (2.5), width 6.8 (3.8), and number of deductions 11.9 (8.0). For the solutions, we observe mean 13.1 (8.2) geometry facts and 2.1 (0.9) area facts.

5 Related Work

(Alvin et al. 2014) first formalized the notion of implicit and explicit atomic geometry facts in a given geometry figure as well as rules for reasoning over those facts. In this paper, we extend that formalism to deal with a richer class of facts involving area facts and rules that relate these facts with each other and also with atomic geometry facts. More significantly, we address the novel challenge of parsing a given coordinate-based geometry image into implicit facts related to both atomic properties and area properties. (Chandrasekaran et al.) proposed an architecture for problem solving with diagrams through different forms of information generation; our algorithms use spatial representation of the given figure and thus follow this architecture. (Matsuzaki et al. 2014) also addressed solution generation for a wide range of mathematics problems including analytic geometry based solely on a textual description. We use a coordinate-based approach and reason about existing figures in our solution generation.

(Seo et al. 2014) describes a technique of diagrammatic understanding as a component of Aristo (Etzioni, Oren and et al. 2015), that extracts implicit, atomic geometry facts from a figure using vision-based techniques. We present a distinct technique to address a more involved problem of also extracting area geometry facts. (Seo et al.) describes a technique for automatically understanding and solving SAT geometry problems while we address the problem of identifying solutions to a shaded area problem (and analyzing those corresponding structures), and not simply the existence of a solution through deduction and logical entailment.

6 Conclusions and Future Work

This paper formally defines a shaded area problem and presents algorithms in a tool *GeoShader* to efficiently solve such problems. Our cross-disciplinary approach combines ideas from computational geometry, logical reasoning, and search heuristics. Future work will involve synthesis of shaded area problems considering existing figures, but should also consider synthesizing new, fresh figures along with problems related to those figures.

References

Alvin, C.; Gulwani, S.; Majumdar, R.; and Mukhopadhyay, S. 2014. Synthesis of geometry proof problems. In *AAAI*.

C. Boyd, e. a. 2006. *Geometry (NJ Edition)*. New York, NY: Glencoe / McGraw-Hill.

CBSE, India. 2012. http://cbse.nic.in/.

Chandrasekaran, B.; Kurup, U.; Banerjee, B.; Josephson, J. R.; and Winkler, R. An architecture for problem solving with diagrams. In *Diagrams, 2004*.

Chew, T. 2008. *Singapore Math Challenge (Grade 5+)*. Frank Schaffer Publications.

Edelsbrunner, H. 1987. *Algorithms in Combinatorial Geometry*. Springer-Verlag.

Etzioni, Oren, and et al. 2015. Ai2 : Aristo, http://allenai.org/aristo.html.

Holt; Rinehart; and Winston. 2007. *Holt Geometry: Homework and Practice Workbook*. Orlando, FL: Holt, Rinehart, and Winston.

Jordan, C. 1893. *Cours D'analyse de L'Ecole Polytechnique*. Gauthier-Villars et fils.

Jurgensen, R.; Brown, R.; and Jurgensen, J. 1988. *Geometry*. Boston, MA: Houghton Mifflin Company.

Larson, R.; Boswell, L.; Kanold, T.; and Stiff, L. 2007. *Geometry*. Evanston, IL: McDougal Littel.

Massachusetts DOE. 2014. http://www.doe.mass.edu/mcas/testitems.html.

Matsuzaki, T.; Iwane, H.; Anai, H.; and Arai, N. H. 2014. The most uncreative examinee: A first step toward wide coverage natural language math problem solving. In *AAAI*.

NY State Education Dept. 2014. http://www.nysedregents.org/regents_math.html.

Seo, M. J.; Hajishirzi, H.; Farhadi, A.; Etzioni, O.; and Malcolm, C. Solving geometry problems: Combining text and diagram interpretation. In *EMNLP, 2015*.

Seo, M. J.; Hajishirzi, H.; Farhadi, A.; and Etzioni, O. 2014. Diagram understanding in geometry questions. In *AAAI*.