Formal Verification of the Empty Hexagon Number

Bernardo Subercaseaux

□

Carnegie Mellon University

Wojciech Nawrocki ⊠®

Carnegie Mellon University

James Gallicchio

□

□

Carnegie Mellon University

Cayden Codel ⊠ ©

Carnegie Mellon University

Mario Carneiro **□ 0**

Carnegie Mellon University

Marijn J. H. Heule ⊠[©]

Carnegie Mellon University

- Abstract

A recent breakthrough in computer-assisted mathematics showed that every set of 30 points in the plane in general position (i.e., no three points on a common line) contains an empty convex hexagon. With a combination of geometric insights and automated reasoning techniques, Heule and Scheucher constructed CNF formulas ϕ_n , with $O(n^4)$ clauses, such that if ϕ_n is unsatisfiable then every set of n points in general position must contain an empty convex hexagon. An unsatisfiability proof for n=30 was then found with a SAT solver using 17 300 CPU hours of parallel computation. In this paper, we formalize and verify this result in the Lean theorem prover. Our formalization covers ideas in discrete computational geometry and SAT encoding techniques by introducing a framework that connects geometric objects to propositional assignments. We see this as a key step towards the formal verification of other SAT-based results in geometry, since the abstractions we use have been successfully applied to similar Erdős-Szekeres-type problems. Overall, we hope that this work sets a new standard for verification when extensive computation is used for discrete geometry problems, and that it increases the trust the mathematical community has in computer-assisted proofs in this area

2012 ACM Subject Classification Theory of computation \rightarrow Logic and verification

Keywords and phrases Empty Hexagon Number, Discrete Computational Geometry, Erdős-Szekeres

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

Mathematicians are often rightfully skeptical of proofs that rely on extensive computation (e.g., the controversy around the four color theorem [39]). Nonetheless, many mathematically-interesting theorems have been resolved that way. SAT solving in particular has been a powerful tool for mathematics, successfully resolving Keller's conjecture [2], the packing chromatic number of the infinite grid [33], the Pythagorean triples problem [18], Lam's problem [3], and one case of the Erdős discrepancy conjecture [23]. All of these proofs rely on the same two-step structure:

- **Reduction)** Show that the mathematical theorem of interest is true if a concrete propositional formula ϕ is unsatisfiable.
- **Solving**) Show that ϕ is indeed unsatisfiable.

Formal methods researchers have devoted significant attention to making the *solving* step reliable, reproducible and trustworthy. Modern SAT solvers produce proofs of unsatisfiability

XX:2 Formal Verification of the Empty Hexagon Number

in formal systems such as DRAT [40] that can in turn be checked with verified proof checkers such as cake_lpr [38]. These tools ensure that when a SAT solver declares a formula ϕ to be unsatisfiable, the formula is indeed unsatisfiable. In contrast, the reduction step can use problem-specific mathematical insights that, when left unverified, threaten the trustworthiness of SAT-based proofs in mathematics. A perfect example of the complexity of this reduction step can be found in a recent breakthrough of Heule and Scheucher [19] in discrete computational geometry. They constructed (and solved) a formula ϕ whose unsatisfiability implies that every set of 30 points, without three in a common line, must contain an empty convex hexagon. However, as is common with such results, their reduction argument was only sketched, relied heavily on intuition, and left several gaps to be filled in.

In this paper we complete and formalize the reduction of Heule and Scheucher in the Lean theorem prover [10]. We do so by connecting existing geometric definitions in the mathematical proof library mathlib [27] to the unsatisfiability of a particular SAT instance, thus setting a new standard for verifying results which rely on extensive computation. Our formalization is publicly available at https://github.com/bsubercaseaux/EmptyHexagonLean/tree/itp2024.

Verification of SAT proofs. Formal verification plays a crucial role in certifying the solving step of SAT-based results. For example, theorem provers and formal methods tools have been used to verify solvers [25, 29, 31] and proof checkers [24, 38]. However, the reduction step has not received similar scrutiny. Some work has been done to verify the reductions to SAT underlying these kinds of mathematical results. The solution to the Pythagorean triples problem was verified in the Coq proof assistant by Cruz-Filipe and coauthors [8,9]. More generally, Giljegård and Wennerbreck [15] provide a CakeML library of verified SAT encodings, which they used to write verified reductions from different puzzles (e.g., Sudoku, Kakuro, the N-queens problem). The reduction verification techniques we use in this paper are based on that of Codel, Avigad, and Heule [6] in the Lean theorem prover.

Formal verification for SAT-based combinatorial geometry was pioneered by Marić [26]. He developed a reduction of a case of the Happy Ending Problem to SAT and formally verified it in Isabelle/HOL. We give a detailed comparison between his work and ours in Section 7.

Lean. Initially developed by Leonardo de Moura in 2013 [10], the Lean theorem prover has become a popular choice for formalizing modern mathematical research. Recent successes include the *Liquid Tensor Experiment* [5] and the proof of the polynomial Freiman-Ruzsa conjecture [16,32], both of which brought significant attention to Lean. A major selling point for Lean is the mathlib project [27], a monolithic formalization of foundational mathematics. By relying on mathlib for definitions, lemmas, and proof tactics, mathematicians can focus on the interesting components of a formalization while avoiding duplication of proof efforts across formalizations. In turn, by making a formalization compatible with mathlib, future proof efforts can rely on work done today. In this spirit, we connect our results to mathlib as much as possible.

The Empty Hexagon Number. In the 1930s, Erdős and Szekeres, inspired by Esther Klein, showed that for any $k \geq 3$, one can find a sufficiently large number n such that every n points in general position (i.e., with no three points collinear) contain a convex k-gon, i.e., a convex polygon with k vertices [12]. The minimal such n is denoted g(k). The same authors later showed that $g(k) > 2^{k-2}$ and conjectured that this bound is tight [11]. Indeed, it is known that g(5) = 9 and g(6) = 17, with the latter result obtained by Szekeres and Peters 71 years after the initial conjecture via exhaustive computer search [37]. Larger cases remain

open, with $g(k) \leq 2^{k+o(k)}$ the best known upper bound [20,35]. This problem is now known as the *Happy Ending Problem*, as it led to the marriage of Klein and Szekeres.

In a similar spirit, Erdős defined h(k) to be the minimal number of points in general position that is guaranteed to contain a k-hole, or $empty\ k$ -gon, meaning a convex k-gon with no other point inside. It is easy to check that h(3) = 3 and h(4) = 5. In 1978, Harborth established that h(5) = 10 [17]. Surprisingly, in 1983, Horton discovered constructions of arbitrarily large point sets that avoid k-holes for $k \geq 7$ [21]. Only h(6) remained. The $Empty\ Hexagon\ Theorem$, establishing h(6) to be finite, was proven independently by Gerken and Nicolás in 2006 [14, 28]. In 2008, Valtr narrowed the range of possible values down to $30 \leq h(6) \leq 1717$, where the problem remained until the breakthrough by Heule and Scheucher [19], who used a SAT solver to prove that $h(6) \leq 30$, a result we refer to as the $Empty\ Hexagon\ Number$.

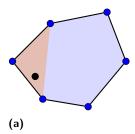
2 Outline of the proof

We will incrementally build sufficient machinery to prove:

▶ **Theorem.** Any finite set of 30 or more points in the plane in general position has a 6-hole.

Outline of the proof. We begin Section 3 with a precise statement in Lean of the above theorem and involved geometric terms. In a nutshell, the proof consists of building a CNF formula ϕ_n such that from any set S of n points in general position without a 6-hole we can construct a satisfying assignment τ_S for ϕ_n . Then, checking that ϕ_{30} is unsatisfiable implies that no such set S of size 30 exists, thus implying the theorem. In order to construct ϕ_n , one must first discretize the continuous space \mathbb{R}^2 . Triple orientations, presented in Section 4, are a way to achieve this. Concretely, any three points p, q, r in general position correspond to either a clockwise turn, denoted by $\sigma(p,q,r) = -1$, or a counterclockwise turn, denoted by $\sigma(p,q,r)=+1$, depending on whether r is above the directed line \overrightarrow{pq} or not. In this way, every set S of points in general position induces an assignment $\sigma_S: S^3 \to \{-1, +1\}$ of triple orientations. We show in Section 4 that whether S contains a k-hole (i.e., HasEmptyKGon k S) depends entirely on σ_S . As each orientation $\sigma(p,q,r)$ can only take two values, we can represent each orientation $\sigma(p,q,r)$ with a boolean variable. Any set of points S in general position thus induces an assignment τ_S over its orientation variables. Because HasEmptyKGon k S depends only on σ_S , it can be written as a boolean formula over the orientation variables. Unfortunately, it is practically infeasible to determine if such a formula is satisfiable with a naïve encoding. In order to create a better encoding, Section 5 shows that one can assume, without loss of generality, that the set of points S is in canonical position. Canonicity eliminates a number of symmetries from the problem – ordering, rotation, and mirroring – significantly reducing the search space. In Section 6, we show the correctness of the efficient encoding of Heule and Scheucher [19] for constructing a smaller CNF formula ϕ_n . Concretely, we show that any finite set of n points in canonical position containing no 6-hole would give rise to a propositional assignment τ_S satisfying ϕ_n . However, ϕ_{30} (depicted in Section 6) is unsatisfiable; therefore no such set of size 30 exists and the theorem follows by contradiction. As detailed in Section 6, to establish unsatisfiability of ϕ_{30} we passed the formula produced by our verified encoder to a SAT solver, and used a verified proof checker to certify the correctness of the resulting unsatisfiability proof. The construction of ϕ_n and τ_S involves sophisticated optimizations which we justify using geometric arguments.

XX:4 Formal Verification of the Empty Hexagon Number



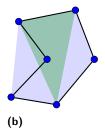


Figure 1 Illustration of the proof for ConvexEmptyIn.iff_triangles. The left subfigure shows how a point in $S \setminus s$ that lies inside s will be inside one of the triangles induced by the convex hull of s (orange triangle). The right subfigure shows how if the ConvexPoints predicate does not hold of s, then some point $a \in s$ will be inside one of the triangles induced by the convex hull of $s \setminus \{a\}$.

3 Geometric Preliminaries

We identify points with elements of \mathbb{R}^2 . Concretely, abbrev Point := EuclideanSpace \mathbb{R} (Fin 2). The next step is to define what it means for a k-gon to be empty (with respect to a set of points) and convex, which we do in terms of mathlib primitives.

```
/-- 'EmptyShapeIn S P' means that 'S' carves out an empty shape in 'P':
the convex hull of 'S' contains no point of 'P' other than those already in 'S'. -/
def EmptyShapeIn (S P : Set Point) : Prop :=
    ∀ p ∈ P \ S, p ∉ convexHull R S

/-- 'ConvexPoints S' means that 'S' consists of extremal points of its convex hull,
i.e., the point set encloses a convex polygon. -/
def ConvexPoints (S : Set Point) : Prop :=
    ∀ a ∈ S, a ∉ convexHull R (S \ {a})

def ConvexEmptyIn (S P : Set Point) : Prop :=
    ConvexPoints S ∧ EmptyShapeIn S P

def HasEmptyKGon (k : Nat) (S : Set Point) : Prop :=
    ∃ s : Finset Point, s.card = k ∧ ↑s ⊆ S ∧ ConvexEmptyIn s S
```

Let ListInGenPos be a predicate that states that a list of points is in *general position*, i.e., no three points lie on a common line (made precise in Section 4). With this we can already state the main theorem of our paper.

At the root of the encoding of Heule and Scheucher is the idea that the ConvexEmptyIn predicate can be determined by analyzing only triangles. In particular, that a set s of k points in a pointset S form an empty convex k-gon if and only if all the $\binom{k}{3}$ triangles induced by vertices in s are empty with respect to S. This is discussed informally in [19, Section 3, Eq. 4]. Concretely, we prove the following theorem:

```
theorem ConvexEmptyIn.iff_triangles {s : Finset Point} {S : Set Point} (sS : \uparrow s \subseteq S) (sz : 3 \le s.card) : ConvexEmptyIn s S \leftrightarrow \forall (t : Finset Point), t.card = 3 \rightarrow t \subseteq s \rightarrow ConvexEmptyIn t S
```

Proof sketch. We first prove a simple monotonicity lemma: if $\mathsf{ConvexPoints}(s)$, then $\mathsf{ConvexPoints}(s')$ for every $s' \subseteq s$, and similarly $\mathsf{EmptyShapeln}(s,S) \Rightarrow \mathsf{EmptyShapeln}(s',S)$ for every set of points S. By instantiating this monotonicity lemma over all subsets $t \subseteq s$ with |t| = 3 we get the forward direction of the theorem. For the backward direction it is easier to reason contrapositively: if the $\mathsf{ConvexPoints}$ predicate does not hold of s, or if s is not empty w.r.t. S, then we want to show that there is a triangle $t \subseteq s$ that is also not empty w.r.t. S. To see this, let H be the convex hull of s, and then by $\mathsf{Carath\'{e}odory}$'s theorem (cf. theorem $\mathsf{convexHull_eq_union}$ from mathlib), every point in H is a convex combination of at most 3 points in s, and consequently, of exactly 3 points in s. If s is non-empty w.r.t. S, then there is a point $p \in S \setminus s$ that belongs to H, and by $\mathsf{Carath\'{e}odory}$, p is a convex combination of 3 points in $s \setminus \{a\}$, and thus lies inside a triangle $t \subseteq s$ (Figure 1a). If s does not hold $\mathsf{ConvexPoints}$, then there is a point s is a convex combination of 3 points in s in s in s in s in s is a convex combination of 3 points in s in s

The next section shows how boolean variables can be used to encode which triangles are empty w.r.t. a pointset, which as the previous theorem shows, can be used to encode the presence or absence of k-holes.

4 Triple Orientations

An essential step for obtaining computational proofs of geometric results is discretization: problems concerning the existence of an object \mathcal{O} in a continuous space such as \mathbb{R}^2 must be reformulated in terms of the existence of a discrete and finitely representable object \mathcal{O}' that a computer can find (or discard its existence). This poses a particular challenge for problems in which the desired geometric object \mathcal{O} is characterized by very specific coordinates of points, requiring to deal with floating point arithmetic or numerical instability. Fortunately, this is not the case for Erdős-Szekeres-type problems such as determining the value of h(k), which are naturally well-suited for computation. This is so because the properties of interest (e.g., convexity, emptiness) can be described in terms of axiomatizable relationships between points and lines (e.g., point p is above the line \overrightarrow{qr} , lines \overrightarrow{qr} and \overrightarrow{st} intersect, etc.), which are invariant under rotations, translations, and even small perturbations of the coordinates. This suggests the problems can be discretized in terms of boolean variables representing these relationships, forgetting the specific coordinates of the points. The combinatorial abstraction that has been most widely used in Erdős-Szekeres-type problems is that of triple orientations [19, 30]. This concept is also known as signotopes [13,34], Knuth's counterclockwise relation [22], or signatures [36]. Given points p, q, r, their triple-orientation is defined as

$$\sigma(p,q,r) = \operatorname{sign} \det \begin{pmatrix} p_x & q_x & r_x \\ p_y & q_y & r_y \\ 1 & 1 & 1 \end{pmatrix} = \begin{cases} 1 & \text{if } p,q,r \text{ are } oriented \text{ counterclockwise,} \\ 0 & \text{if } p,q,r \text{ are collinear,} \\ -1 & \text{if } p,q,r \text{ are } oriented \text{ clockwise.} \end{cases}.$$

An example is illustrated in Figure 2. We directly use mathlib's definition of the determinant to define σ .

```
inductive Orientation : Type where
| cw -- clockwise := -
| ccw -- counter clockwise := +
| collinear -- := 0
```

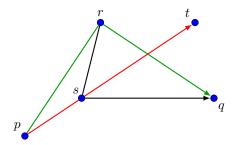


Figure 2 Illustration of triple orientations, where $\sigma(p, r, q) = -1$, $\sigma(r, s, q) = 1$, and $\sigma(p, s, t) = 0$.

```
noncomputable def \sigma (p q r : Point) : Orientation := let det := Matrix.det !![p.x, q.x, r.x ; p.y, q.y, r.y ; 1, 1, 1] if 0 < det then ccw else if det < 0 then cw else collinear
```

Using the function σ we can define the notion of general position for collections (e.g., finite sets, lists, etc.) of points, simply postulating that $\sigma(p,q,r) \neq 0$ for every three distinct points p,q,r in the collection. Furthermore, we can start formalizing sets of points that are equivalent with respect to their triple orientations, and consequently, properties of pointsets that are fully captured by their triple orientations (orientation properties).

```
structure \sigmaEquiv (S T : Set Point) where f : Point \to Point bij : Set.BijOn f S T parity : Bool -- See Section 4 for details on this field \sigma_-eq : \forall (p \in S) (q \in S) (r \in S), \sigma p q r = parity ^{\frown} \sigma (f p) (f q) (f r) def OrientationProperty (P : Set Point \to Prop) := \forall {{S T}}, S \simeq \sigma T \to P S \to P T - '\simeq \sigma' is infix notation for '\sigmaEquiv'
```

To illustrate how these notions will be used, let us consider the property $\pi_k(S) \triangleq$ "pointset S contains an empty convex k-gon", formalized as HasEmptyKGon.

Based on ConvexEmptyIn.iff_triangles, we know that $\pi_k(S)$ can be written in terms of whether certain triangles are empty w.r.t S. We can define triangle membership using σ , and prove its equivalence to the geometric definition.

```
/-- 'Means that 'a' is in the triangle 'pqr', possibly on the boundary. -/
def PtInTriangle (a : Point) (p q r : Point) : Prop :=
    a ∈ convexHull R {p, q, r}

/-- 'Means that 'a' is in the triangle 'pqr' strictly, not on the boundary. -/
def σPtInTriangle (a p q r : Point) : Prop :=
    σ p q a = σ p q r ∧ σ p a r = σ p q r ∧ σ a q r = σ p q r

theorem σPtInTriangle_iff {a p q r : Point} (gp : InGenPos4 a p q r) :
    σPtInTriangle a p q r ↔ PtInTriangle a p q r
```

Heule and Scheucher used the orientation-based definition [19], and as it is common in the area, its equivalence to the *ground-truth* mathematical definition was left implicit. This equivalence, formalized in theorem σ PtInTriangle_iff is not trivial to prove: the forward

direction in particular requires reasoning about convex combinations and determinants. Using the previous theorem, we can generalize to k-gons as follows.

```
def \sigmaIsEmptyTriangleFor (a b c : Point) (S : Set Point) : Prop := \forall s \in S, \neg \sigmaPtInTriangle s a b c def \sigmaHasEmptyKGon (n : Nat) (S : Set Point) : Prop := \exists s : Finset Point, s.card = n \land \uparrows \subseteq S \land \forall (a \in s) (b \in s) (c \in s), a \neq b \rightarrow a \neq c \rightarrow b \neq c \rightarrow \sigmaIsEmptyTriangleFor a b c S theorem \sigmaHasEmptyKGon_iff_HasEmptyKGon (gp : ListInGenPos pts) : \sigmaHasEmptyKGon n pts.toFinset \leftrightarrow HasEmptyKGon n pts.toFinset

Then, because \sigmaHasEmptyKGon is ultimately defined in terms of \sigma, we can prove lemma OrientationProperty_\sigmaHasEmptyKGon : OrientationProperty (\sigmaHasEmptyKGon n)

Which in combination with theorem \sigmaHasEmptyKGon_iff_HasEmptyKGon, provides
```

Let us discuss why the previous theorem is relevant, as it plays an important role in the

theorem OrientationProperty_HasEmptyKGon : OrientationProperty (HasEmptyKGon n)

formalization of Erdős-Szekeres-type problems. This boils down to two reasons:

- 1. If we prove that the function σ is invariant under a certain transformation of its arguments (e.g., rotations, translations, etc.) then we can directly conclude that any orientation property is invariant under the same transformation. This is a powerful tool for applying manipulations to pointsets that preserve the properties of interest, which will be key for symmetry breaking (see Section 5). For a concrete example, consider a proof of an Erdős-Szekeres-type result that starts by saying "we assume without loss of generality that points p_1, \ldots, p_n all have positive y-coordinates". As translations are σ -equivalences, we can see that this assumption indeed does not impact the truth of any orientation property.
- 2. As introduced at the beginning of this section, SAT encodings for Erdős-Szekeres-type problems are based on capturing properties like convexity or emptiness in terms of triple orientations, thus reducing a continuous search space to a discrete one. Because we have proved that $\pi_k(S)$ is an orientation property, the values of σ for all triples of points in S contain enough information to determine whether $\pi_k(S)$ or not. Therefore, we have proved that given n points it is enough to analyze the values of σ over these points, a discrete space with at most 2^{n^3} possibilities, instead of $(\mathbb{R}^2)^n$. This is the key idea that will allow us to transition from the finitely-verifiable statement "no set of triple orientations over n points satisfies property π_k " to "no set of n points satisfies property π_k ".

4.1 Properties of orientations

We now prove, assuming points are sorted left-to-right (which is justified in Section 5), that certain σ -implication-properties hold. Consider four points p,q,r,s with $p_x < q_x < r_x < s_x$. If p,q,r are oriented counterclockwise, and q,r,s are oriented counterclockwise as well, then it follows that p,r,s must be oriented counterclockwise (see Figure 3). We prove a number of properties of this form:

```
theorem \sigma_prop<sub>1</sub> (h : Sorted<sub>4</sub> p q r s) (gp : InGenPos<sub>4</sub> p q r s) : \sigma p q r = ccw \rightarrow \sigma q r s = ccw \rightarrow \sigma p r s = ccw
```

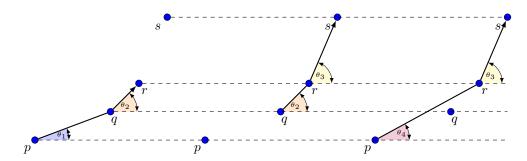


Figure 3 Illustration for $\sigma(p, q, r) = 1 \wedge \sigma(q, r, s) = 1 \implies \sigma(p, r, s) = 1$. As we have assumptions $\theta_3 > \theta_2 > \theta_4$ by the forward direction of the *slope-orientation equivalence*, we deduce $\theta_3 > \theta_4$, and then conclude $\sigma(p, r, s) = 1$ by the backward direction of the *slope-orientation equivalence*.

[...]

```
theorem \sigma_prop_3 (h : Sorted_4 p q r s) (gp : InGenPos_4 p q r s) : \sigma p q r = cw \rightarrow \sigma q r s = cw \rightarrow \sigma p r s = cw
```

They will be used in justifying the addition of clauses (5) and (6); clauses like these or the one below are easily added, and are commonly used to reduce the search space in SAT encodings [19,30,34,36].

$$(\neg o_{a,b,c} \lor \neg o_{a,c,d} \lor o_{a,b,d}) \land (o_{a,b,c} \lor o_{a,c,d} \lor \neg o_{a,b,d})$$

$$\tag{1}$$

Our proofs of these properties are based on an equivalence between the orientation of a triple of points and the *slopes* of the lines that connect them. Namely, if p,q,r are sorted from left to right, then (i) $\sigma(p,q,r)=1 \iff \mathsf{slope}(\overrightarrow{pq}) < \mathsf{slope}(\overrightarrow{pr})$ and (ii) $\sigma(p,q,r)=1 \iff \mathsf{slope}(\overrightarrow{pr}) < \mathsf{slope}(\overrightarrow{qr})$. By first proving these *slope-orientation* equivalences we can then easily prove $\sigma_{\mathtt{prop}_1}$ and others, as illustrated in Figure 3.

5 Symmetry Breaking

Symmetry breaking plays a key role in modern SAT-solving by substantially reducing the search space of assignments to a formula [1,7]. For example, if one proves that all satisfying assignments to a formula ϕ have either (i) $x_1=0, x_2=1$, or (ii) $x_1=1, x_2=0$, and there is a bijection between satisfying assignments of form (i) and satisfying assignments of form (ii), then one can assume, without loss of generality, that $x_1=0, x_2=1$, and thus add unit clauses $\overline{x_1}$ and x_2 to the formula ϕ while preserving its satisfiability.

In the context of the Empty Hexagon Number, the symmetry breaking done by Heule and Scheucher consists in assuming that in order to search for a list of 30 points in general position without a 6-hole, it suffices to can search only amongst lists of 30 points in *canonical* position. These are defined as follows.

- ▶ **Definition 1** (Canonical Position). A list $L = (p_1, ..., p_n)$ of points is said to be in canonical position if it satisfies all the following properties:
- **(x-order)** The points are sorted with respect to their x-coordinates, i.e., $x(p_i) < x(p_j)$ for all $1 \le i < j \le n$.

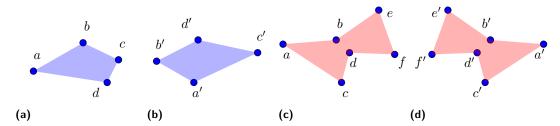


Figure 4 The pointsets depicted in Figures 4a and 4b are σ -equivalent with parity := false since the bijection f defined by $(a,b,c,d) \mapsto (b',d',c',a')$ satisfies $\sigma(p_i,p_j,p_k) = \sigma(f(p_i),f(p_j),f(p_j))$ for every $\{p_i,p_j,p_k\} \subseteq \{a,b,c,d\}$. On the other hand, no orientation-preserving bijection exists for Figures 4c and 4d, which are only σ -equivalent with parity := true.

- (General Position) No three points are collinear, i.e., for all $1 \le i < j < k \le n$, we have $\sigma(p_i, p_j, p_k) \ne 0$.
- **(CCW-order)** All orientations $\sigma(p_1, p_i, p_j)$, with $1 < i < j \le n$, are counterclockwise.
- (Lex order) The list of orientations $\left(\sigma\left(p_{\lceil\frac{n}{2}\rceil-1},p_{\lceil\frac{n}{2}\rceil},p_{\lceil\frac{n}{2}\rceil+1}\right),\ldots,\sigma\left(p_2,p_3,p_4\right)\right)$ is not lexicographically smaller than the list $\left(\sigma\left(p_{\lfloor\frac{n}{2}\rfloor+1},p_{\lfloor\frac{n}{2}\rfloor+2},p_{\lfloor\frac{n}{2}\rfloor+3}\right),\ldots,\sigma\left(p_{n-2},p_{n-1},p_n\right)\right)$.

This symmetry breaking assumption not only reduces the search space of the SAT solver, but it is required for the correctness of the encoding, as clauses (2)–(6) rely on points being sorted from left to right. Before discussing the proof of correctness of symmetry breaking, let us first focus on the last condition, expressed explicitly in clause (7) of the encoding. The main idea behind this condition is that reflecting a pointset, i.e., applying the map $(x,y)\mapsto (-x,y)$, preserves the presence of k-holes, or convex k-gons. This is the reason for incorporating the parity flag in the definition of σ -equivalence. As illustrated in Figure 4, there are pointsets that are only σ -equivalent to their reflections with parity := true. We are now ready to state the main symmetry breaking theorem and sketch its proof.

```
theorem symmetry_breaking : ListInGenPos 1 \to \exists w : CanonicalPoints, Nonempty (1.toFinset \simeq \sigma w.points.toFinset)
```

Proof Sketch. The proof proceeds in 6 steps, illustrated in Figure 5. In each of the steps, we will construct a new list of points that is σ -equivalent to the previous one, and the last one will be in canonical position.¹ The main justification for each step is that, given that the function σ is defined as a sign of the determinant, applying transformations that preserve (or, when parity := true, uniformly reverse) the sign of the determinant will preserve (or uniformly reverse) the values of σ . In particular, given the identity $\det(AB) = \det(A) \det(B)$, if we apply a transformation to the points that corresponds to multiplying by a matrix B such that $\det(B) > 0$, then $\operatorname{sign}(\det(A)) = \operatorname{sign}(\det(AB))$, and thus orientations will be preserved. In step 1, we transform the list of points so that no two points share the same x-coordinate. This can be done by applying a rotation to the list of points, which corresponds to multiplying by a rotation matrix. Rotations always have determinant 1. In step 2, we translate all points by a constant vector t, by multiplying by a translation matrix, so that the left most point gets position (0,0), and naturally every other point will have

Even though we defined σ -equivalence for sets of points, our formalization goes back and forth between sets and lists. Given that symmetry breaking distinguishes between the order of the points e.g., x-order, this proof proceeds over lists. All permutations of a list are immediately σ -equivalent.

XX:10 Formal Verification of the Empty Hexagon Number

 $(x,y) \mapsto (y/x,1/x).$

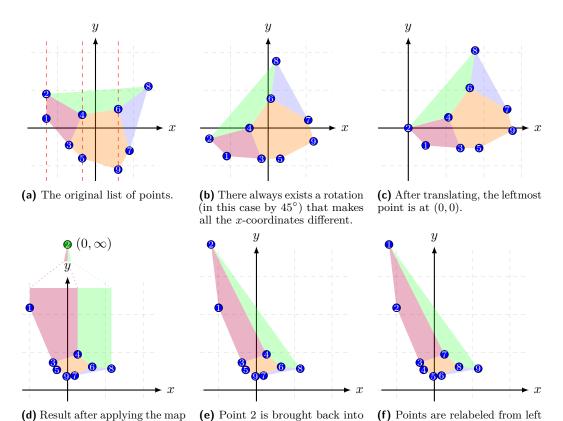


Figure 5 Illustration of the proof of the symmetry breaking theorem. Note that the highlighted holes are preserved as σ -equivalence is preserved. For simplicity we have omitted the illustration of the Lex order property.

to right.

the real plane.

a positive x-coordinate. Let L_2 be the list of points after this transformation, excluding (0,0) which we will denote by p_1 . Then, in step 3, we apply the projective transformation $f:(x,y)\mapsto (y/x,1/x)$ to every point in L_2 , showing that this preserves orientations within L_2 . To see that this mapping is a σ -equivalence consider that

$$\begin{aligned} & \operatorname{sign} \det \begin{pmatrix} p_x & q_x & r_x \\ p_y & q_y & r_y \\ 1 & 1 & 1 \end{pmatrix} = \operatorname{sign} \det \begin{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} p_y/p_x & q_y/q_x & r_y/r_x \\ 1/p_x & 1/q_x & 1/r_x \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} p_x & 0 & 0 \\ 0 & q_x & 0 \\ 0 & 0 & r_x \end{pmatrix} \end{pmatrix} \\ & = \operatorname{sign} \begin{pmatrix} 1 \cdot \det \begin{pmatrix} p_y/p_x & q_y/q_x & r_y/r_x \\ 1/p_x & 1/q_x & 1/r_x \\ 1 & 1 & 1 \end{pmatrix} \cdot p_x q_x r_x \end{pmatrix} = \operatorname{sign} \det \begin{pmatrix} p_y/p_x & q_y/q_x & r_y/r_x \\ 1/p_x & 1/q_x & 1/r_x \\ 1 & 1 & 1 \end{pmatrix}. \end{aligned}$$

To preserve orientations with respect to the leftmost point (0,0), we set $f((0,0))=(0,\infty)$, a special point that is treated separately as follows. As the function σ takes points in \mathbb{R}^2 as arguments, we need to define an extension $\sigma_{(0,\infty)}(q,r)=\begin{cases} 1 & \text{if } q_x < r_x \\ -1 & \text{otherwise.} \end{cases}$, We then show that $\sigma((0,0),q,r)=\sigma_{(0,\infty)}(f(q),f(r))$ for all points $q,r\in L_2$. In step 4, we sort the list L_2 by x-coordinate in increasing order, thus obtaining a list L_3 . This can be done while preserving σ -equivalence because sorting corresponds to a permutation, and all permutations of a list are σ -equivalent by definition. In step 5, we check whether the Lex order condition above is

satisfied in L_3 , and if it is not, we reflect the pointset, which as explained above, preserves σ -equivalence by leveraging the parity option in the definition. Note that in such a case we need to relabel the points from left to right again. In step 6, we bring point $(0, \infty)$ back into the range by first finding a constant c such that all points in L_3 are to the right of the line y = c, and then finding a large enough value M such that (c, M) has the same orientation with respect to the other points as $(0, \infty)$ did, meaning that $\sigma((c, M), q, r) = \sigma_{(0, \infty)}(q, r)$ for every $q, r \in L_3$. Finally, we note that the list of points obtained in step 6 satisfies the CCW-order property by the following reasoning: if $1 < i < j \le n$ are indices, then

$$\begin{split} \sigma(p_1,p_i,p_j) &= 1 \iff \sigma((c,M),p_i,p_j) = 1 \\ &\iff \sigma_{(0,\infty)}(p_i,p_j) = 1 \\ &\iff (p_i)_x < (p_j)_x \end{split} \qquad \text{(By definition of } \sigma_{(0,\infty)}) \\ &\iff \mathsf{true}. \qquad \text{(By step 4, since points are sorted and } i < j) \end{split}$$

This concludes the proof.

6 The Encoding and Its Correctness

Having established the reduction to orientations, and the symmetry-breaking assumption of canonicity, we now turn to the construction of a CNF formula ϕ_n whose unsatisfiability would imply that every set of n points contains a 6-hole.² The formula is detailed in Section 6.

Variables. Let $S = (p_1, \ldots, p_n)$ be the list of points in canonical position. We explain the variables of ϕ_n by specifying their values in the propositional assignment τ_S that is our intended model of ϕ_n corresponding to S. We then have:

- For every $2 \le a < b < c \le n$, $o_{a,b,c}$ is true iff $\sigma(p_a,p_b,p_c) = +1.^3$ The first optimization observes that orientations are antisymmetric: if (p,q,r) is counter-clockwise then (q,p,r) is clockwise, etc. Thus one only needs $o_{a,b,c}$ for ordered triples (a,b,c), reducing the number of orientation variables by a factor of 3! = 6 relative to using all triples. The second optimization uses the **CCW-order** property of canonical positions: since all $o_{1,a,b}$ are true, we may as well omit them from the encoding.
- Next, for every a < b < c with a < i < b or b < i < c, the variable $c_{i;a,b,c}$ is true iff σ PtInTriangle S[i] S[a] S[b] S[c] holds. By σ PtInTriangle_iff, this is true exactly iff p_i is inside the triangle $p_ap_bp_c$. The reason for assuming (a,b,c) to be ordered is again symmetry: $p_ap_bp_c$ is the same triangle as $p_ap_cp_b$, etc. Furthermore thanks to the x-order property of canonical positions, if p_i is in the triangle then $x(p_a) < x(p_i) < x(p_c)$. This implies that a < i < c, leaving one case distinction permuting (i,b).
- For every a < b < c, $h_{a,b,c}$ is true iff σ IsEmptyTriangleFor S[a] S[b] S[c] S holds. By a geometro-combinatorial connection analogous to ones above, this is true iff $p_a p_b p_c$ is a 3-hole.
- Finally, one defines 4-cap, 5-cap, and 4-cup variables. For a+1 < c < d, $\mathsf{u}_{a,c,d}^4$ is true iff there is b with a < b < c with $\sigma(p_a, p_b, p_c) = \sigma(p_b, p_c, p_d) = -1$. $\mathsf{v}_{a,c,d}^4$ is analogous, except in that the two orientations are required to be counterclockwise. These are the 4-caps and 4-cups, respectively. The 5-cap variables $\mathsf{u}_{a,d,e}^5$ are defined for a+2 < d < e.

² Satisfiability of ϕ_n would not necessarily imply the existence of a point set without a 6-hole, due to the realizability problem (see e.g., [34]).

³ Since the point set is in general position, we have $\neg o_{a,b,c} \iff \sigma(p_a,p_b,p_c) = -1$.

XX:12 Formal Verification of the Empty Hexagon Number

$$c_{i;a,b,c} \to ((o_{a,b,c} \leftrightarrow o_{a,i,c}) \land (o_{a,b,c} \leftrightarrow \overline{o_{a,i,b}})) \text{ for all } 2 \le a < i < b < c \le n$$

$$c_{i;a,b,c} \to ((o_{a,b,c} \leftrightarrow o_{a,i,c}) \land (o_{a,b,c} \leftrightarrow \overline{o_{b,i,c}})) \text{ for all } 2 \le a < b < i < c \le n$$
(3)

$$\left(\bigwedge_{\substack{a < i < c \\ i \neq b}} \overline{\mathsf{c}_{i;a,b,c}} \right) \to \mathsf{h}_{a,b,c} \quad \text{ for all } 2 \le a < b < c \le n$$
 (4)

$$o_{a,b,c} \land o_{a,c,d} \rightarrow o_{a,b,d}$$
 for all $2 \le a < b < c < d \le n$ (5)

$$\overline{\mathsf{o}_{a,b,c}} \wedge \overline{\mathsf{o}_{a,c,d}} \to \overline{\mathsf{o}_{a,b,d}} \quad \text{for all } 2 \le a < b < c < d \le n$$

$$\left(\mathsf{o}_{\lceil\frac{n}{2}\rceil-1,\lceil\frac{n}{2}\rceil,\lceil\frac{n}{2}\rceil+1},\ldots,\mathsf{o}_{2,3,4}\right)\succeq_{\mathrm{lex}}\left(\mathsf{o}_{\lfloor\frac{n}{2}\rfloor+1,\lfloor\frac{n}{2}\rfloor+2,\lfloor\frac{n}{2}\rfloor+3},\ldots,\mathsf{o}_{n-2,n-1,n}\right)$$

$$(7)$$

$$\overline{\mathsf{o}_{a,b,c}} \wedge \overline{\mathsf{o}_{b,c,d}} \to \mathsf{u}_{a,c,d}^4 \quad \text{ for all } 2 \le a < b < c < d \le n$$

$$\mathbf{o}_{a,b,c} \wedge \mathbf{o}_{b,c,d} \to \mathbf{v}_{a,c,d}^4 \quad \text{for all } 2 \le a < b < c < d \le n$$

$$\mathbf{u}_{a,b,c}^4 \wedge \overline{\mathbf{o}_{b,c,d}} \wedge \mathbf{h}_{a,b,d} \to \mathbf{u}_{a,c,d}^5 \quad \text{for all } 2 \le a < b < c < d \le n, \ a+1 < b$$
(10)

$$\mathsf{u}_{a,b,c}^4 \wedge \overline{\mathsf{o}_{b,c,d}} \wedge \mathsf{h}_{a,b,d} \to \mathsf{u}_{a,c,d}^5 \quad \text{for all } 2 \le a < b < c < d \le n, \ a+1 < b \tag{10}$$

$$\mathsf{u}_{a,c,d}^4 \to \overline{\mathsf{o}_{a,c,d}} \quad \text{ for all } 2 \le a < c < d \le n, \ a+1 < c \tag{11}$$

$$\mathsf{v}_{a,c,d}^4 \to \mathsf{o}_{a,c,d}$$
 for all $2 \le a < c < d \le n, \ a+1 < c$ (12)

$$\neg (\mathsf{u}_{a,d,e}^5 \land \mathsf{o}_{a,p,e}) \quad \text{ for all } 2 \le a < d < e \le n, \ a < p < e, \ a+2 < d$$
 (13)

$$\neg (\mathsf{u}_{a,d,e}^{5} \wedge \overline{\mathsf{o}_{d,e,f}}) \quad \text{for all } 2 \le a < d < e < f \le n, \ a+2 < d$$

$$\neg (\mathsf{u}_{a,c,d}^{4} \wedge \mathsf{v}_{a,c',d}^{4} \wedge \mathsf{h}_{a,c,c'}) \quad \text{for all } 2 \le a < c < c' < d \le n, \ a+1 < c$$

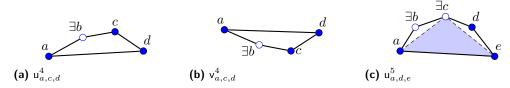
$$\neg (\mathsf{u}_{a,c,d}^{4} \wedge \mathsf{v}_{a,c',d}^{4} \wedge \mathsf{h}_{a,c',c}) \quad \text{for all } 2 \le a < c' < c < d \le n, \ a+1 < c'$$
(15)

$$\neg (\mathsf{u}_{a,c,d}^4 \wedge \mathsf{v}_{a,c',d}^4 \wedge \mathsf{h}_{a,c,c'}) \quad \text{ for all } 2 \le a < c < c' < d \le n, \ a+1 < c \tag{15}$$

$$\neg (\mathsf{u}_{a,c,d}^4 \land \mathsf{v}_{a,c',d}^4 \land \mathsf{h}_{a,c',c}) \quad \text{for all } 2 \le a < c' < c < d \le n, \ a+1 < c'$$
 (16)

$$\neg(\mathsf{v}_{a,c,d}^4 \land \mathsf{o}_{c,d,e} \land \mathsf{h}_{a,c,e}) \quad \text{ for all } 2 \le a < c < d < e \le n, \ a+1 < c \tag{17}$$

Figure 6 Encoding based on that of Heule and Scheucher for the Empty Hexagon Number [19]. Each line determines a set of clauses. Unsatisfiability of the formula below for n=30 implies $h(6) \leq 30$, as detailed throughout the paper.



■ Figure 7 Illustration of the 4-cap (7a), 4-cup (7b), and 5-cap (7c) variables. The highlighted region denotes an empty triangle.

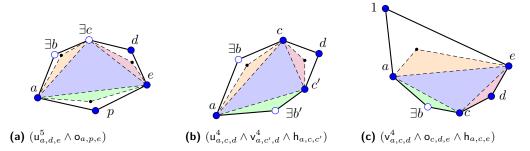


Figure 8 Illustration of some *forbidden configurations* that imply 6-holes. Figure 8a corresponds to the configuration forbidden by clause (13), Figure 8b to the one forbidden by clause (15), and Figure 8c to clause (17). All highlighted regions denote empty triangles.

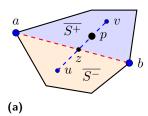
We set $\mathfrak{u}_{a,d,e}^5$ to true iff there exists c with a+1 < c < d such that $\mathfrak{u}_{a,c,d}^4$, $\mathfrak{o}_{c,d,e}$, and $\mathfrak{h}_{a,c,e}$ are all true. Intuitively, 4-caps and 4-cups are clockwise and counterclockwise arcs of length 4, respectively, whereas 5-caps are clockwise arcs of length 5 containing a 3-hole. All three are depicted in Figure 7. The usage of these variables is crucial to an efficient encoding: we will show below that a hexagon can be covered by only 4 triangles, so one need not consider all $\binom{6}{3}$ triangles contained within it.

Satisfaction. We now have to justify that the clauses of ϕ_n are satisfied by the intended interpretation τ_S for a 6-hole-free point set S. The variable-defining clauses (2)–(4) and (8)–(12) follow essentially by definition combined with boolean reasoning. The orientation properties (5) and (6) have been established in the family of theorems $\sigma_{\tt prop}_i$. The lexicographic ordering clauses (7) follow from the Lex order property of canonical positions. Thus we are left with clauses (13)–(17) which forbid the presence of certain 6-holes.⁴ We illustrate why clause (13) is true. The contrapositive is easier to state: if τ_S satisfies $\mathsf{u}_{a,d,e}^5 \wedge \mathsf{o}_{a,p,e}$, then S contains a 6-hole. The intuitive argument is depicted in Figure 8a. The clause directly implies the existence of a convex hexagon apedcb such that ace is a 3-hole. It turns out that this is enough to ensure the existence of a 6-hole by "flattening" the triangles ape, edc, and cba, if necessary, to obtain empty triangles ap'e, ed'c, and cb'a, which can be assembled into a 6-hole ap'ed'cb'.

Justifying this formally turned out to be complex, requring a fair bit of reasoning about point Arcs and σ CCWPoints: lists of points winding around a convex polygon. Luckily, the main argument can be summarized in terms of two facts: (a) any triangle abc contains an

⁴ They are intended to forbid all 6-holes, but proving completeness is not necessary for an unsatisfiability-based result.

XX:14 Formal Verification of the Empty Hexagon Number



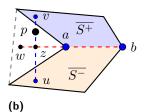


Figure 9 Illustration of the proof for split_convexHull. (a) Given point p, we obtain points u and v inside the two halves and z as the point of intersection with the line \overline{ab} . (b) In this (contradictory) situation, the point z has ended up outside the segment \overline{ab} , because S is not actually convex. In this case we construct w such that z is on the \overline{wa} segment, and observe that w, z, a, b are collinear.

empty triangle ab'c; and (b) empty shapes sharing a common line segment can be glued together. Formally, (a) can be stated as

```
theorem \sigmaIsEmptyTriangleFor_exists (gp : ListInGenPos S) (abc : [a, b, c] \subseteq S) : \exists b' \in S, \sigma a b' c = \sigma a b c \land (b' = b \lor \sigmaPtInTriangle b' a b c) \land \sigmaIsEmptyTriangleFor a b' c S.toFinset
```

Proof. Given points p, q, say that $p \leq q$ iff p is in the triangle aqc. This is a preorder. Now, the set $S' = \{x \in S \mid \sigma(a, x, c) = \sigma(a, b, c) \land x \leq b\}$ is finite and so has a weakly minimal element b', in the sense that no $x \in S'$ has x < b'. Emptiness of ab'c follows by minimality.

Moving on, (b) follows from a *triangulation lemma*: given any convex point set S and a line \overrightarrow{ab} between two vertices of S, the convex hull of S is contained in the convex hulls of points on either side of \overrightarrow{ab} . That is:

```
theorem split_convexHull (cvx : ConvexPoints S) : \forall {a b}, a \in S \rightarrow b \in S \rightarrow convexHull \mathbb R S \subseteq convexHull \mathbb R {x \in S | \sigma a b x \neq ccw} \cup convexHull \mathbb R {x \in S | \sigma a b x \neq ccw}
```

Proof. Let $S^+ = \{x \in S \mid \sigma(a,b,x) \geq 0\}$ and $S^- = \{x \in S \mid \sigma(a,b,x) \leq 0\}$ be the two sets in the theorem, and let $p \in \overline{S}$, where \overline{S} denotes the convex hull of S. Assume WLOG that $\sigma(a,b,p) \geq 0$. (We would like to show that $p \in \overline{S^+}$.) Now p is a convex combination of elements of S^+ and elements of S^- , so there exist points $u \in \overline{S^-}$ and $v \in \overline{S^+}$ such that p lies on the \overline{uv} line. Because $\{x \mid \det(a,b,x) \leq 0\} \supseteq S^-$ is convex, it follows that $\det(a,b,u) \leq 0$, and likewise $\det(a,b,v) \geq 0$, so they lie on opposite sides of the \overline{ab} line and hence \overline{uv} intersects \overline{ab} at a point z. The key point is that z must in fact be on the line segment \overline{ab} ; assuming that this was the case, we could obtain z as a convex combination of a and b, and b as a convex combination of b and b, and b as a convex combination of b and b, suppose not, so that b lies between b and b (see Figure 9b). (The case where b is on the b side is similar.) We can decompose b as a convex combination of some b and b, which means that b and b are collinear and appear in this order on the line. Therefore a is a convex combination of b and b, which means that b and b which violates convexity of b.

By contraposition, the triangulation lemma directly implies that if $\{x \in S \mid \sigma(a, b, x) \neq +1\}$ and $\{x \in S \mid \sigma(a, b, x) \neq -1\}$ are both empty shapes in P, then S is an empty shape in P.

Running the CNF. Having now shown that our main result follows if ϕ_{30} is unsatisfiable, we run a distributed computation to check its unsatisfiability. We solve the SAT formula ϕ_{30} produced by Lean using the same setup as Heule and Scheucher [19], although using different hardware: the Bridges 2 cluster of the Pittsburgh Supercomputing Center [4]. Following Heule and Scheucher, we partition the problem into 312418 subproblems. Each of these subproblems was solved using CaDiCaL version 1.9.5. The solver produced an LRAT proof for each execution, which was validated using the cake_lpr verified checker on-the-fly in order to avoid writing/storing/reading large files. The total runtime was 25876.5 CPU hours, or roughly 3 CPU years. The difference in runtime relative to Heule and Scheucher's original run is purely due to the difference in hardware. Additionally, we validated that the subproblems cover the entire search space as Heule and Scheucher did [19, Section 7.3]. This was done by verifying the unsatisfiability of another formula that took 20 seconds to solve.

7 Related Work

Our formalization is closely related to a prior development in which Marić put proofs of $g(6) \leq 17$ on a more solid foundation [26]. The inequality, originally obtained by Szekeres and Peters [37] using a specialized, unverified search algorithm, was confirmed by Marić using a formally-verified SAT encoding. Marić introduced an optimized encoding based on nested convex hull structures, which, when combined with performance advances in modern SAT solvers, significantly improved the search time over the unverified computation.

Our work focuses on the closely-related problem of determining k-hole numbers h(k). Rather than devise a new SAT encoding, we use essentially the same encoding presented by Heule and Scheucher [19]. Interestingly, a (verified) proof of $g(6) \leq 17$ can be obtained as a corollary of our development. We can assert the hole variables $h_{a,b,c}$ as true while leaving the remainder of the encoding in Section 6 unchanged, which trivializes constraints about emptiness so that only the convexity constraints remain.⁵ The resulting CNF formula asserts the existence of a set of n points with no convex 6-gon. We checked this formula to be unsatisfiable for n = 17, giving the same result as Marić:

```
theorem gon_6_theorem (pts : List Point) (gp : ListInGenPos pts) (h : pts.length \geq 17) : HasConvexKGon 6 pts.toFinset
```

Since both formalizations can be executed, we performed a direct comparison against Marić's encoding. On a personal laptop, we found that it takes negligible time (below 1s) for our verified Lean encoder to output the full CNF. In contrast, Marić's encoder, extracted from Isabelle/HOL code, took 437s to output a CNF (this was compiled on Isabelle/HOL 2016, the latest version that accepts the codebase without broader changes). To circumvent the encoder slowness, Marić wrote a C++ encoder whose code was manually compared against the Isabelle/HOL specification. We do not need to resort to an unverified implementation.

As for the encodings, ours took 28s to solve, while the Marić encoding took 787s (both using cadical). This difference is likely accounted for in the relative size of the encodings, in particular their symmetry breaking strategies. For k = 6 and n points, the encoding of Heule and Scheucher uses $O(n^4)$ clauses, whereas the one of Marić uses $O(n^6)$ clauses. They are based on different ideas: the former as detailed in Section 5, whereas the latter on nested convex hulls. The different approaches have been discussed by Scheucher [30]. This progress

⁵ This modification was performed by an author who did not understand this part of the proof, nevertheless having full confidence in its correctness thanks to the Lean kernel having checked every assertion.

XX:16 Formal Verification of the Empty Hexagon Number

in solve times represents an encouraging state of affairs; we are optimistic that if continued, it could lead to an eventual resolution of q(7).

Further differences include what exactly was formally proven. As with most work in this area, we use the combinatorial abstraction of triple orientations. We and Marić alike show that point sets in \mathbb{R}^2 satisfy orientation properties (Section 4). However, our work goes further in building the connection between geometry and combinatorics: our definitions of convexity and emptiness (Section 3), and consequently the theorem statements, are geometric ones based on convex hulls as defined in Lean's mathlib [27]. In contrast, Marić axiomatizes these properties in terms of σ . A skeptical reviewer must manually verify that these combinatorial definitions correspond to the desired geometric concept.

A final point of difference concerns the verification of SAT proofs. Marić fully reconstructs some of the SAT proofs on which his results depend, though not the main one for g(6), in an NbE-based proof checker for Isabelle/HOL. We make no such attempt for the time being, instead passing our SAT proofs through the formally verified proof checker cake_lpr [38] and asserting unsatisfiability of the CNF as an axiom in Lean. Thus we trust that the CNF formula produced by the verified Lean encoder is the same one whose unsatisfiability was checked by cake_lpr.

8 Concluding Remarks

We have proved the correctness of the main result of Heule and Scheucher [19], implying $h(6) \leq 30$. Given that the lower bound h(6) > 29 can be checked directly (see [19]), we conclude the result h(6) = 30 is indeed correct. We believe this work puts a happy ending to one line of research started by Klein, Erdős and Szekeres in the 1930s. Prior to formalization, the result of Heule and Scheucher relied on the correctness of various components of a highly sophisticated encoding that are hard to validate manually. We developed a significant theory of combinatorial geometry that was not present in mathlib. Beyond the main theorem presented here, we showed how our framework can be used for other related theorems such as g(6) = 17, and we hope it can be used for proving many further results in the area.

Our formalization required approximately 300 hours of work over 3 months by researchers with significant experience formalizing mathematics in Lean. The final version of our proofs consists of approximately 4.7k lines of Lean code; about 26% are lemmas that should be moved to upstream libraries, about 40% develops the theory of orientations in plane geometry, and the remaining 34% (1550 LOC) validates the symmetry breaking and SAT encoding.

We substantially simplified the symmetry-breaking argument presented by Heule and Scheucher, and derived in turn from Scheucher [30]. Moreover, we found a small error in their proof, as their transformation uses the mapping $(x,y) \mapsto (x/y,-1/y)$, and incorrectly assumes that x/y is increasing for points in CCW-order, whereas only the slopes y/x are increasing. Similarly, we found a typo in the statement of the Lex order condition that did not match the (correct) code of Heule and Scheucher. Our formalization corrects this.

In terms of future work, we hope to formally verify the result $h(7) = \infty$ due to Horton [21], and other results in Erdős-Szekeres style problems. A key challenge for the community is to improve the connection between verified SAT tools and ITPs. This presents a significant engineering task for proofs that are hundreds of terabytes long (as in this result). Although we are confident that our results are correct, the trust story at this connection point has room for improvement.

References

1 A. Biere, M. Heule, H. van Maaren, and T. Walsh. *Handbook of Satisfiability: Volume 185 Frontiers in Artificial Intelligence and Applications*. IOS Press, NLD, 2009.

- 2 Joshua Brakensiek, Marijn Heule, John Mackey, and David Narváez. The Resolution of Keller's Conjecture, 2023. arXiv:1910.03740.
- 3 Curtis Bright, Kevin K. H. Cheung, Brett Stevens, Ilias S. Kotsireas, and Vijay Ganesh. A SAT-based resolution of Lam's Problem. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, AAAI 2021, pages 3669–3676. AAAI Press, 2021. URL: https://doi.org/10.1609/aaai.v35i5.16483, doi:10.1609/AAAI.V35I5.16483.
- 4 Shawn T. Brown, Paola Buitrago, Edward Hanna, Sergiu Sanielevici, Robin Scibek, and Nicholas A. Nystrom. *Bridges-2: A Platform for Rapidly-Evolving and Data Intensive Research*, pages 1–4. Association for Computing Machinery, New York, NY, USA, 2021.
- 5 Davide Castelvecchi. Mathematicians welcome computer-assisted proof in 'grand unification' theory. *Nature*, 595(7865):18–19, June 2021. URL: http://dx.doi.org/10.1038/d41586-021-01627-2, doi:10.1038/d41586-021-01627-2.
- 6 Cayden Codel, Marijn J. H. Heule, and Jeremy Avigad. Verified Encodings for SAT Solvers. In Alexander Nadel and Kristin Yvonne Rozier, editors, *Proceedings of the 23rd conference on Formal Methods In Computer-Aided Design*, 2023.
- 7 James Crawford, Matthew Ginsberg, Eugene Luks, and Amitabha Roy. Symmetry-breaking predicates for search problems. In *Proc. KR'96*, 5th Int. Conf. on Knowledge Representation and Reasoning, pages 148–159. Morgan Kaufmann, 1996.
- 8 Luís Cruz-Filipe, João Marques-Silva, and Peter Schneider-Kamp. Formally Verifying the Solution to the Boolean Pythagorean Triples Problem. *J. Autom. Reason.*, 63(3):695–722, oct 2019. doi:10.1007/s10817-018-9490-4.
- 9 Luís Cruz-Filipe and Peter Schneider-Kamp. Formally Proving the Boolean Pythagorean Triples Conjecture. In Thomas Eiter and David Sands, editors, LPAR-21. 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning, volume 46 of EPiC Series in Computing, pages 509-522. EasyChair, 2017. URL: https://easychair.org/publications/paper/xq6J, doi:10.29007/jvdj.
- 10 Leonardo de Moura, Soonho Kong, Jeremy Avigad, Floris van Doorn, and Jakob von Raumer. The Lean Theorem Prover (System Description). In Amy P. Felty and Aart Middeldorp, editors, Automated Deduction - CADE-25, pages 378–388, Cham, 2015. Springer International Publishing.
- 11 Paul Erdős and George Szekeres. On some extremum problems in elementary geometry. *Ann. Univ. Sci. Budapest. Eötvös Sect. Math.*, 3(4):53–62, 1960.
- Paul Erdős and György Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935. URL: http://eudml.org/doc/88611.
- Stefan Felsner and Helmut Weil. Sweeps, arrangements and signotopes. *Discrete Applied Mathematics*, 109(1):67–94, April 2001. doi:10.1016/S0166-218X(00)00232-8.
- Tobias Gerken. Empty Convex Hexagons in Planar Point Sets. Discrete & Computational Geometry, 39(1):239–272, mar 2008. doi:10.1007/s00454-007-9018-x.
- Sofia Giljegård and Johan Wennerbeck. Puzzle Solving with Proof. Master's thesis, Chalmers University of Technology, 2021.
- W. T. Gowers, Ben Green, Freddie Manners, and Terence Tao. On a conjecture of Marton, 2023. arXiv:2311.05762.
- 17 Heiko Harborth. Konvexe Fünfecke in ebenen Punktmengen. *Elemente der Mathematik*, 33:116–118, 1978. URL: http://eudml.org/doc/141217.
- Marijn J. H. Heule, Oliver Kullmann, and Victor W. Marek. Solving and Verifying the Boolean Pythagorean Triples Problem via Cube-and-Conquer, page 228-245. Springer International Publishing, 2016. URL: http://dx.doi.org/10.1007/978-3-319-40970-2_15, doi:10.1007/978-3-319-40970-2_15.

XX:18 Formal Verification of the Empty Hexagon Number

- Marijn J. H. Heule and Manfred Scheucher. Happy ending: An empty hexagon in every set of 30 points, 2024. arXiv:2403.00737.
- 20 Andreas F Holmsen, Hossein Nassajian Mojarrad, János Pach, and Gábor Tardos. Two extensions of the erdős-szekeres problem. arXiv preprint arXiv:1710.11415, 2017.
- J. D. Horton. Sets with No Empty Convex 7-Gons. Canadian Mathematical Bulletin, 26(4):482–484, 1983. doi:10.4153/CMB-1983-077-8.
- Donald E. Knuth. Axioms and Hulls. In Donald E. Knuth, editor, Axioms and Hulls, Lecture Notes in Computer Science, pages 1–98. Springer, Berlin, Heidelberg, 1992. doi: 10.1007/3-540-55611-7_1.
- 23 Boris Konev and Alexei Lisitsa. A SAT Attack on the Erdos Discrepancy Conjecture, 2014. arXiv:1402.2184.
- Peter Lammich. Efficient Verified (UN)SAT Certificate Checking. *Journal of Automated Reasoning*, 64(3):513–532, March 2020. doi:10.1007/s10817-019-09525-z.
- Filip Maric. Formal verification of a modern SAT solver by shallow embedding into Isabelle/HOL. Theor. Comput. Sci., 411(50):4333-4356, 2010. URL: https://doi.org/10.1016/j.tcs.2010.09.014, doi:10.1016/J.TCS.2010.09.014.
- 26 Filip Maric. Fast formal proof of the Erdős-Szekeres conjecture for convex polygons with at most 6 points. J. Autom. Reason., 62(3):301–329, 2019. URL: https://doi.org/10.1007/s10817-017-9423-7, doi:10.1007/S10817-017-9423-7.
- The mathlib Community. The Lean mathematical library. In Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, POPL '20. ACM, January 2020. URL: http://dx.doi.org/10.1145/3372885.3373824, doi:10.1145/3372885.3373824.
- 28 Carlos M. Nicolas. The Empty Hexagon Theorem. Discrete & Computational Geometry, 38(2):389–397, September 2007. doi:10.1007/s00454-007-1343-6.
- 29 Duckki Oe, Aaron Stump, Corey Oliver, and Kevin Clancy. Versat: A Verified Modern SAT Solver. In Viktor Kuncak and Andrey Rybalchenko, editors, Verification, Model Checking, and Abstract Interpretation, pages 363–378, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- Manfred Scheucher. Two disjoint 5-holes in point sets. Computational Geometry, 91:101670, December 2020. doi:10.1016/j.comgeo.2020.101670.
- 31 Sarek Høverstad Skotåm. CreuSAT, Using Rust and Creusot to create the world's fastest deductively verified SAT solver. Master's thesis, University of Oslo, 2022. URL: https://www.duo.uio.no/handle/10852/96757.
- 32 Leila Sloman. 'A-Team' of Math Proves a Critical Link Between Addition and Sets. https://www.quantamagazine.org/a-team-of-math-proves-a-critical-link-between-addition-and-sets-20231206/, December 2023.
- 33 Bernardo Subercaseaux and Marijn J. H. Heule. The Packing Chromatic Number of the Infinite Square Grid is 15. In Sriram Sankaranarayanan and Natasha Sharygina, editors, Tools and Algorithms for the Construction and Analysis of Systems 29th International Conference, TACAS 2023, Held as Part of ETAPS 2022, Proceedings, Part I, volume 13993 of Lecture Notes in Computer Science, page 389–406. Springer, 2023. doi:10.1007/978-3-031-30823-9_20.
- Bernardo Subercaseaux, John Mackey, Marijn J. H. Heule, and Ruben Martins. Minimizing pentagons in the plane through automated reasoning, 2023. arXiv:2311.03645.
- 35 Andrew Suk. On the erdős-szekeres convex polygon problem. *Journal of the American Mathematical Society*, 30(4):1047–1053, 2017.
- 36 George Szekeres and Lindsay Peters. Computer solution to the 17-point Erdős-Szekeres problem. The ANZIAM Journal, 48(2):151–164, 2006. doi:10.1017/S144618110000300X.
- 37 George Szekeres and Lindsay Peters. Computer solution to the 17-point erdős-szekeres problem. The ANZIAM Journal, 48(2):151–164, 2006.
- Yong Kiam Tan, Marijn J. H. Heule, and Magnus O. Myreen. Verified Propagation Redundancy and Compositional UNSAT Checking in CakeML. *International Journal on Software Tools* for Technology Transfer, 25(2):167–184, April 2023. doi:10.1007/s10009-022-00690-y.

Mark Walters. It Appears That Four Colors Suffice: A Historical Overview of the Four-Color Theorem. 2004. URL: https://api.semanticscholar.org/CorpusID:14382286.

40 Nathan Wetzler, Marijn J. H. Heule, and Warren A. Hunt. DRAT-trim: Efficient checking and trimming using expressive clausal proofs. In Carsten Sinz and Uwe Egly, editors, *Theory and Applications of Satisfiability Testing – SAT 2014*, pages 422–429, Cham, 2014. Springer International Publishing.