Formal Verification of the Empty Hexagon Number

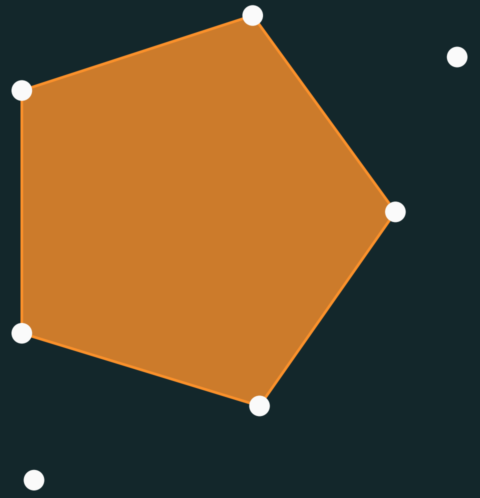
Bernardo Subercaseaux[1], Wojciech Nawrocki[1], James Gallicchio[1],
Cayden Codel[1], Mario Carneiro[1], Marijn J. H. Heule[1]

Interactive Theorem Proving | September 9th, 2024
Tbilisi, Georgia

[1] Carnegie Mellon University, USA

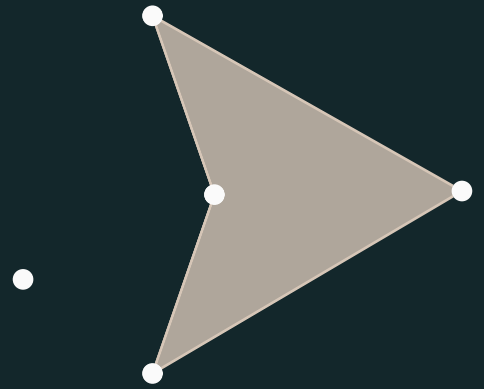
Empty k-gons

Fix a set S of points on the plane, *no three collinear*. A **k-hole** is a convex k-gon with no point of S in its interior.



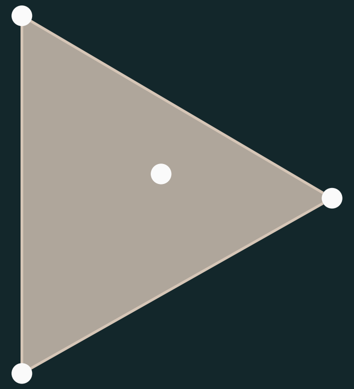
5-hole ✓

convex 5-gon ✓



4-hole ✗

convex 4-gon ✗



3-hole ✗

convex 3-gon ✓

5 points must contain a 4-hole

Theorem (Klein 1932). Every set of 5 points in the plane contains a 4-hole.

5 points must contain a 4-hole

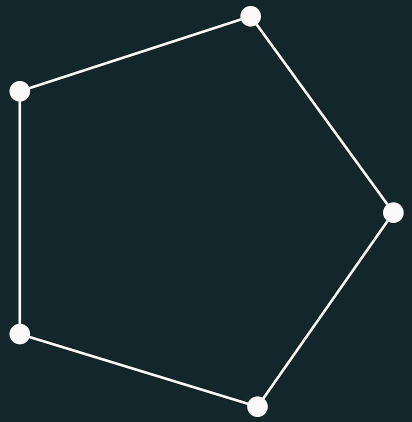
Theorem (Klein 1932). Every set of 5 points in the plane contains a 4-hole.

Proof. By cases on the size of the convex hull.

5 points must contain a 4-hole

Theorem (Klein 1932). Every set of 5 points in the plane contains a 4-hole.

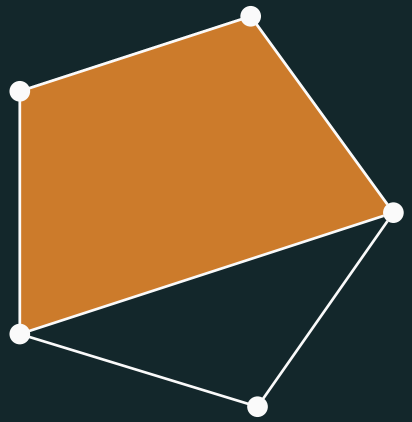
Proof. By cases on the size of the convex hull.



5 points must contain a 4-hole

Theorem (Klein 1932). Every set of 5 points in the plane contains a 4-hole.

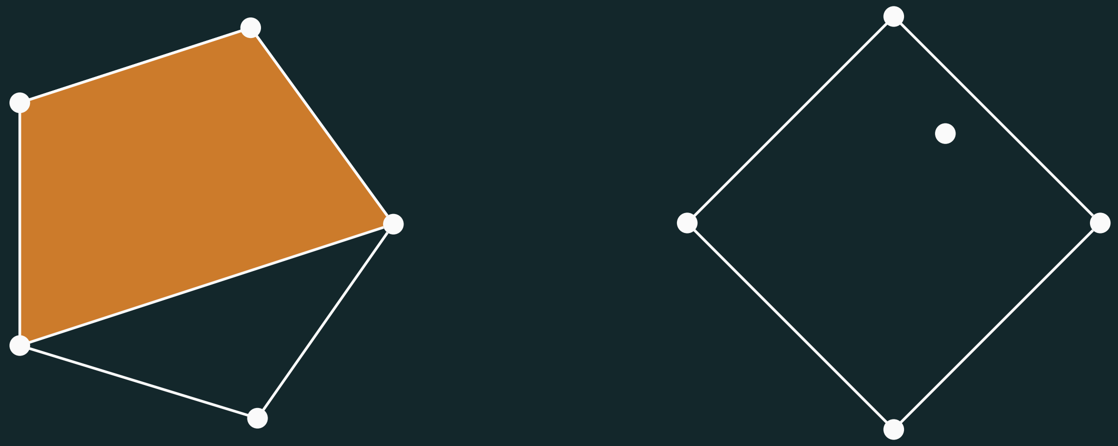
Proof. By cases on the size of the convex hull.



5 points must contain a 4-hole

Theorem (Klein 1932). Every set of 5 points in the plane contains a 4-hole.

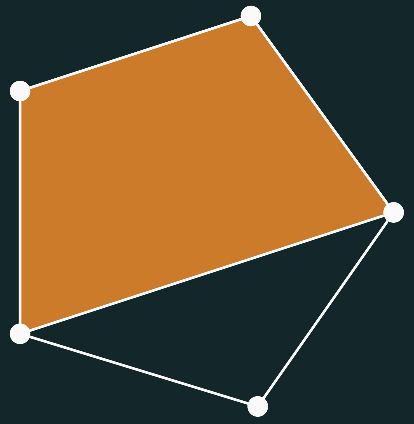
Proof. By cases on the size of the convex hull.

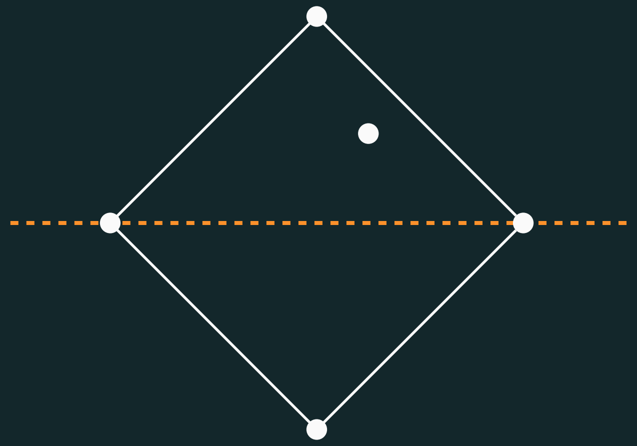


5 points must contain a 4-hole

Theorem (Klein 1932). Every set of 5 points in the plane contains a 4-hole.

Proof. By cases on the size of the convex hull.

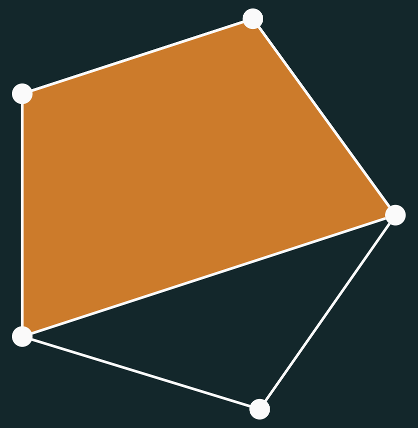


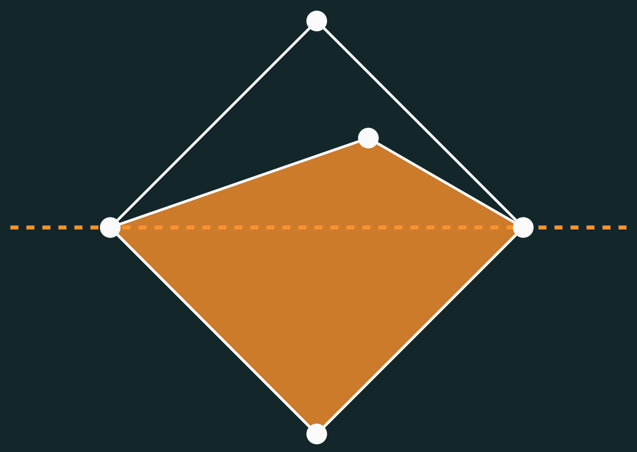


5 points must contain a 4-hole

Theorem (Klein 1932). Every set of 5 points in the plane contains a 4-hole.

Proof. By cases on the size of the convex hull.

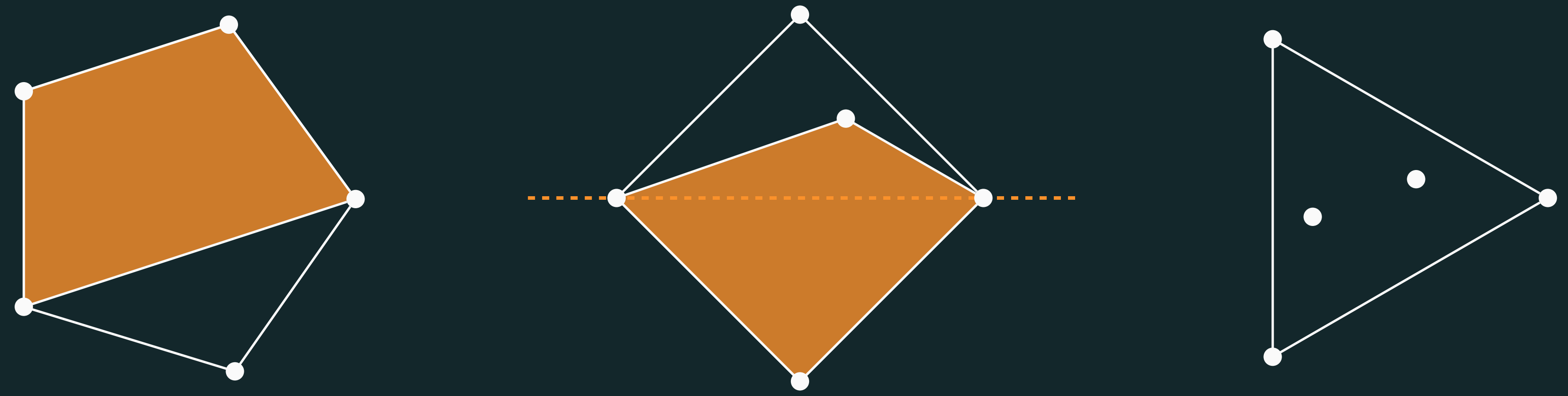




5 points must contain a 4-hole

Theorem (Klein 1932). Every set of 5 points in the plane contains a 4-hole.

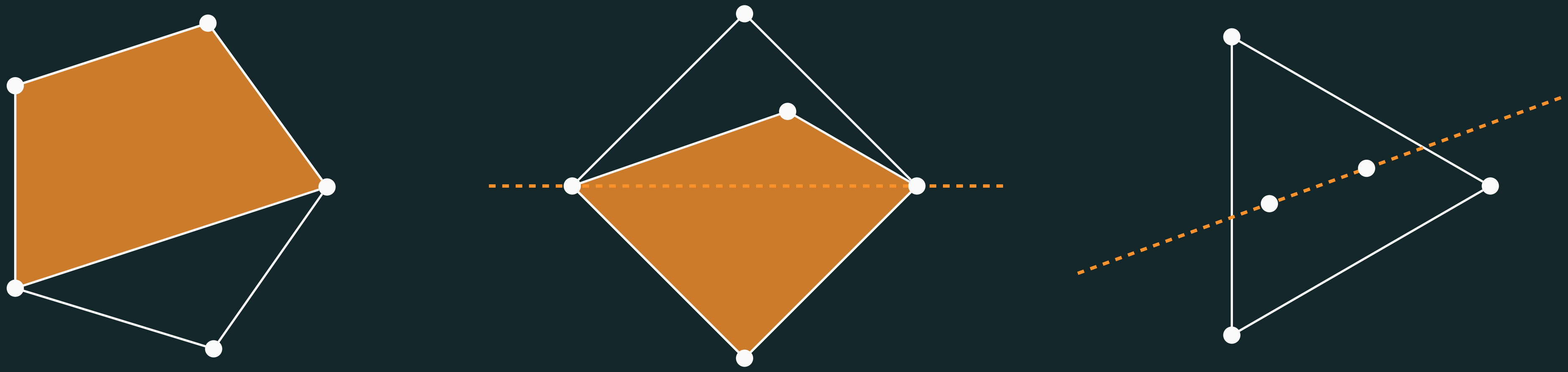
Proof. By cases on the size of the convex hull.



5 points must contain a 4-hole

Theorem (Klein 1932). Every set of 5 points in the plane contains a 4-hole.

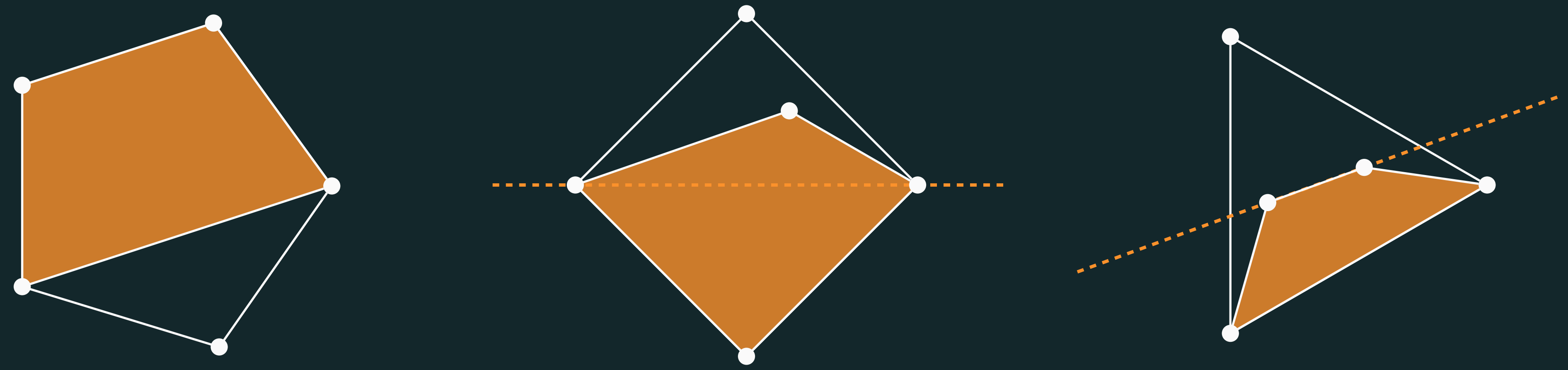
Proof. By cases on the size of the convex hull.



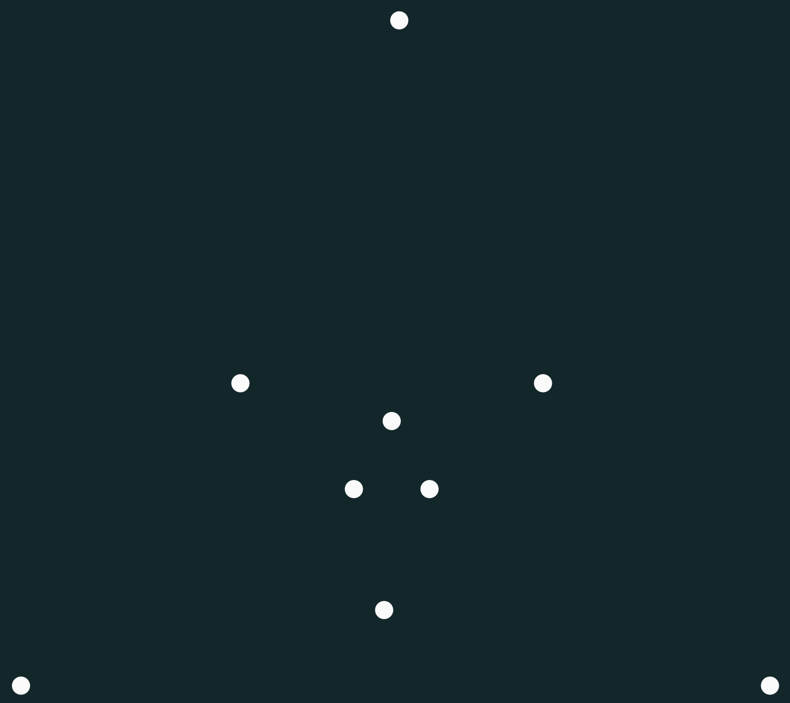
5 points must contain a 4-hole

Theorem (Klein 1932). Every set of 5 points in the plane contains a 4-hole.

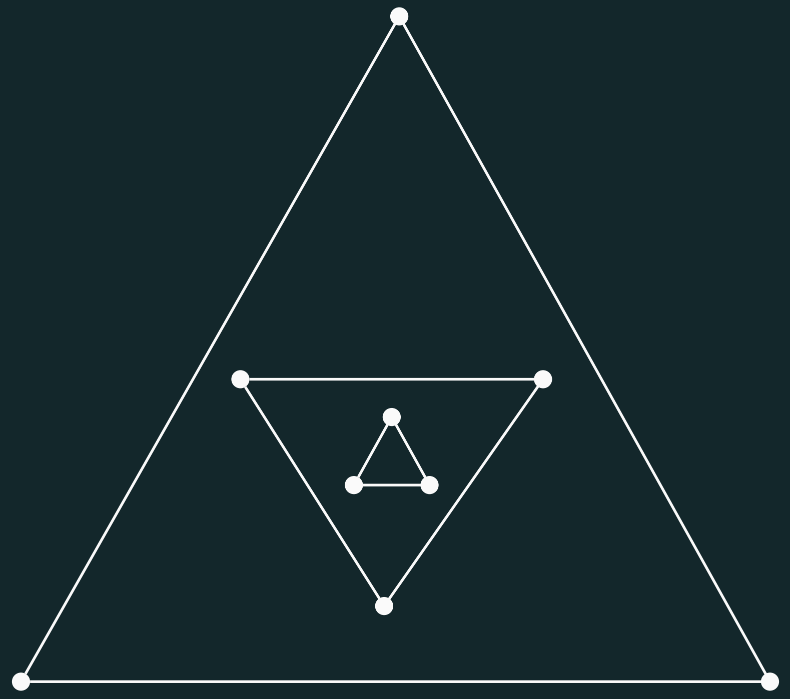
Proof. By cases on the size of the convex hull.



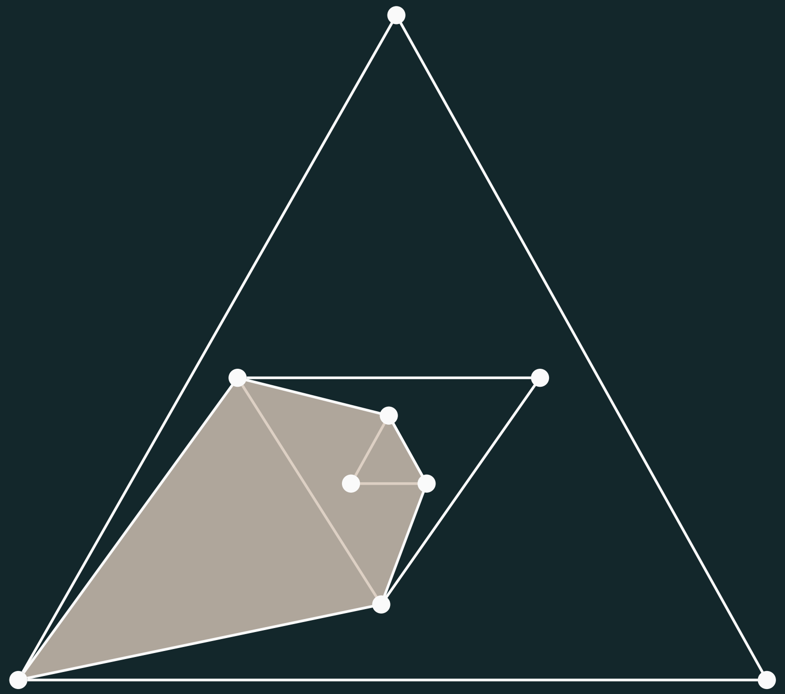
9 points with no 5-holes



9 points with no 5-holes

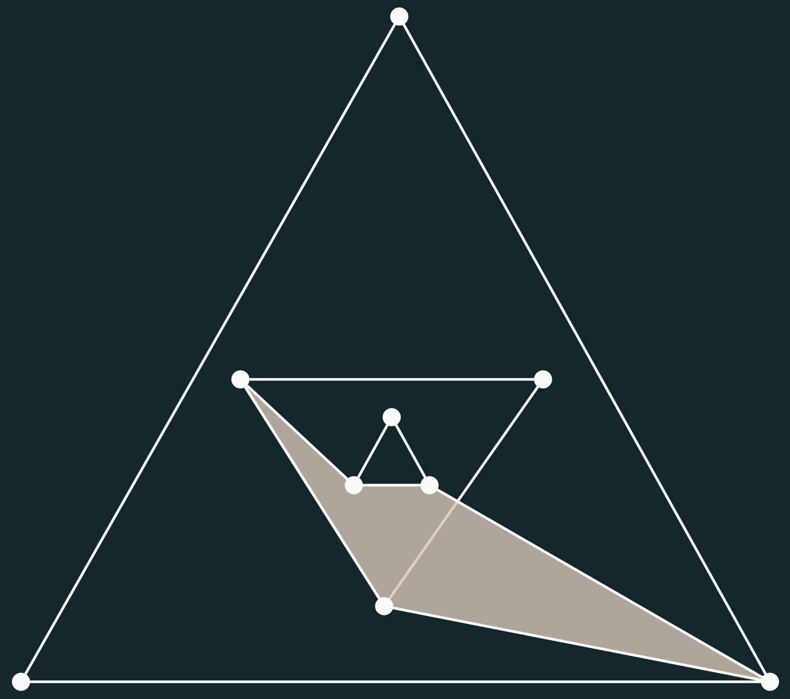


9 points with no 5-holes



5-hole ✘
convex 5-gon ✓

9 points with no 5-holes



convex 5-gon ✘

The Happy Ending Problem

$g(k)$ = least n s.t. any set of n points must contain a **convex k-gon**

$h(k)$ = least n s.t. any set of n points must contain a **k-hole**

We just showed $h(4) \leq 5$ and $9 < h(5)$

Theorem (Erdős and Szekeres 1935). For a fixed k, every *sufficiently large* set of points contains a convex k-gon. So all $g(k)$ are finite.

Theorem (Horton 1983). For any $k \geq 7$, there exist arbitrarily large sets of points containing no k-holes. So $h(7) = h(8) = \ldots = \infty$.

The Happy Ending Problem

$g(k)$ = least n s.t. any set of n points must contain a **convex k-gon**

$h(k)$ = least n s.t. any set of n points must contain a **k-hole**

We just showed $h(4) \leq 5$ and $9 < h(5)$

Theorem (Erdős and Szekeres 1935). For a fixed k, every *sufficiently large* set of points contains a convex k-gon. So all $g(k)$ are finite.

Theorem (Horton 1983). For any $k \geq 7$, there exist arbitrarily large sets of points containing no k-holes. So $h(7) = h(8) = \ldots = \infty$.

The Happy Ending Problem

$g(k) =$ least n s.t. any set of n points must contain a **convex k-gon**

$h(k) =$ least n s.t. any set of n points must contain a **k-hole**

We just showed $h(4) \leq 5$ and $9 < h(5)$

Theorem (Erdős and Szekeres 1935). For a fixed k, every *sufficiently large* set of points contains a convex k-gon. So all $g(k)$ are finite.

Theorem (Horton 1983). For any $k \geq 7$, there exist arbitrarily large sets of points containing no k-holes. So $h(7) = h(8) = \ldots = \infty$.

The Happy Ending Problem

$g(k)$ = least n s.t. any set of n points must contain a **convex *k*-gon**

$h(k)$ = least n s.t. any set of n points must contain a ***k*-hole**

We just showed $h(4) \leq 5$ and $9 < h(5)$

Theorem (Erdős and Szekeres 1935). For a fixed k, every *sufficiently large* set of points contains a convex k-gon. So all $g(k)$ are finite.

Theorem (Horton 1983). For any $k \geq 7$, there exist arbitrarily large sets of points containing no k-holes. So $h(7) = h(8) = \ldots = \infty$.

The Happy Ending Problem

$g(k)$ = least n s.t. any set of n points must contain a **convex k-gon**

$h(k)$ = least n s.t. any set of n points must contain a **k-hole**

We just showed $h(4) \leq 5$ and $9 < h(5)$

Theorem (Erdős and Szekeres 1935). For a fixed k, every *sufficiently large* set of points contains a convex k-gon. So all $g(k)$ are finite.

Theorem (Horton 1983). For any $k \geq 7$, there exist arbitrarily large sets of points containing no k-holes. So $h(7) = h(8) = \ldots = \infty$.

Known tight bounds

$h(3) = 3$ (trivial)
$h(4) = 5$ (Klein 1932)
$h(5) = 10$ (Harborth 1978)
$h(6) = 30$ **(Overmars 2003; Heule and Scheucher 2024)**

$g(3) = 3$ (trivial)
$g(4) = 5$ (Klein 1932)
$g(5) = 9$ (Makai 1930s)
$g(6) = 17$ (Szekeres and Peters 2006)

We formally verified all the above in Lean.

Upper bounds by combinatorial reduction to SAT.

- We focused on $h(6)$, established this year.
- $g(6)$ previously verified in Isabelle/HOL (Marić 2019).
- Efficient SAT encoding of Heule & Scheucher speeds up $g(6)$ verification.

Lower bounds by checking concrete sets of points.

Known tight bounds

$h(3) \leq 3$ (trivial)
$h(4) \leq 5$ (Klein 1932)
$h(5) \leq 10$ (Harborth 1978)
$h(6) \leq 30$ **(Overmars 2003; Heule and Scheucher 2024)**

$g(3) \leq 3$ (trivial)
$g(4) \leq 5$ (Klein 1932)
$g(5) \leq 9$ (Makai 1930s)
$g(6) \leq 17$ (Szekeres and Peters 2006)

We formally verified all the above in Lean.

Upper bounds by combinatorial reduction to SAT.

- We focused on $h(6)$, established this year.
- $g(6)$ previously verified in Isabelle/HOL (Marić 2019).
- Efficient SAT encoding of Heule & Scheucher speeds up $g(6)$ verification.

Lower bounds by checking concrete sets of points.

Known tight bounds

$h(3) \leq 3$ (trivial)
$h(4) \leq 5$ (Klein 1932)
$h(5) \leq 10$ (Harborth 1978)
$h(6) \leq 30$ **(Overmars 2003; Heule and Scheucher 2024)**

$g(3) \leq 3$ (trivial)
$g(4) \leq 5$ (Klein 1932)
$g(5) \leq 9$ (Makai 1930s)
$g(6) \leq 17$ (Szekeres and Peters 2006)

We formally verified all the above in Lean.

Upper bounds by combinatorial reduction to SAT.

- We focused on $h(6)$, established this year.
- $g(6)$ previously verified in Isabelle/HOL (Marić 2019).
- Efficient SAT encoding of Heule & Scheucher speeds up $g(6)$ verification.

Lower bounds by checking concrete sets of points.

Known tight bounds

$h(3) \leq 3$ (trivial)
$h(4) \leq 5$ (Klein 1932)
$h(5) \leq 10$ (Harborth 1978)
$h(6) \leq 30$ **(Overmars 2003; Heule and Scheucher 2024)**

$g(3) \leq 3$ (trivial)
$g(4) \leq 5$ (Klein 1932)
$g(5) \leq 9$ (Makai 1930s)
$g(6) \leq 17$ (Szekeres and Peters 2006)

We formally verified all the above in Lean.

Upper bounds by combinatorial reduction to SAT.

- We focused on $h(6)$, established this year.
- $g(6)$ previously verified in Isabelle/HOL (Marić 2019).
- Efficient SAT encoding of Heule & Scheucher speeds up $g(6)$ verification.

Lower bounds by checking concrete sets of points.

Known tight bounds

$2 < h(3)$ (trivial)
$4 < h(4)$ (Klein 1932)
$9 < h(5)$ (Harborth 1978)
$29 < h(6)$ **(Overmars 2003; Heule and Scheucher 2024)**

$2 < g(3)$ (trivial)
$4 < g(4)$ (Klein 1932)
$8 < g(5)$ (Makai 1930s)
$16 < g(6)$ (Szekeres and Peters 2006)

We formally verified all the above in Lean.

Upper bounds by combinatorial reduction to SAT.

- We focused on $h(6)$, established this year.
- $g(6)$ previously verified in Isabelle/HOL (Marić 2019).
- Efficient SAT encoding of Heule & Scheucher speeds up $g(6)$ verification.

Lower bounds by checking concrete sets of points.

Known tight bounds

$h(3) = 3$ (trivial)
$h(4) = 5$ (Klein 1932)
$h(5) = 10$ (Harborth 1978)
$h(6) = 30$ **(Overmars 2003; Heule and Scheucher 2024)**

$g(3) = 3$ (trivial)
$g(4) = 5$ (Klein 1932)
$g(5) = 9$ (Makai 1930s)
$g(6) = 17$ (Szekeres and Peters 2006)

We formally verified all the above in Lean.

Upper bounds by combinatorial reduction to SAT.

- We focused on $h(6)$, established this year.
- $g(6)$ previously verified in Isabelle/HOL (Marić 2019).
- Efficient SAT encoding of Heule & Scheucher speeds up $g(6)$ verification.

Lower bounds by checking concrete sets of points.

SAT-solving mathematics

Reduction. Show that a mathematical theorem is true if a propositional formula φ is unsatisfiable.

Solving. Show that φ is indeed unsatisfiable using a SAT solver.

- Solving is reliable, reproducible, and trustworthy: formal proof systems (DRAT) and verified proof checkers (`cake_lpr`).
- But reduction is problem-specific, and involves complex transformations: *focus of our work.*

SAT-solving mathematics

Reduction. Show that a mathematical theorem is true if a propositional formula φ is unsatisfiable.

Solving. Show that φ is indeed unsatisfiable using a SAT solver.

- Solving is reliable, reproducible, and trustworthy: formal proof systems (DRAT) and verified proof checkers (`cake_lpr`).

- But reduction is problem-specific, and involves complex transformations: *focus of our work*.

SAT-solving mathematics

Reduction. Show that a mathematical theorem is true if a propositional formula φ is unsatisfiable.

Solving. Show that φ is indeed unsatisfiable using a SAT solver.

- Solving is reliable, reproducible, and trustworthy: formal proof systems (DRAT) and verified proof checkers (`cake_lpr`).

- But reduction is problem-specific, and involves complex transformations: *focus of our work.*

SAT-solving mathematics

Reduction. Show that a mathematical theorem is true if a propositional formula φ is unsatisfiable.

Solving. Show that φ is indeed unsatisfiable using a SAT solver.

- Solving is reliable, reproducible, and trustworthy: formal proof systems (DRAT) and verified proof checkers (`cake_lpr`).
- But reduction is problem-specific, and involves complex transformations: *focus of our work.*

Reduction from geometry to SAT

1. Discretize from continuous coordinates in $\mathbb{R}^2$ to boolean variables.

2. Points can be put in *canonical form* without removing k-holes.

```
theorem symmetry_breaking {l : List Point} :
  3 ≤ l.length → PointsInGenPos l →
  ∃ w : CanonicalPoints, l ≼σ w.points
```

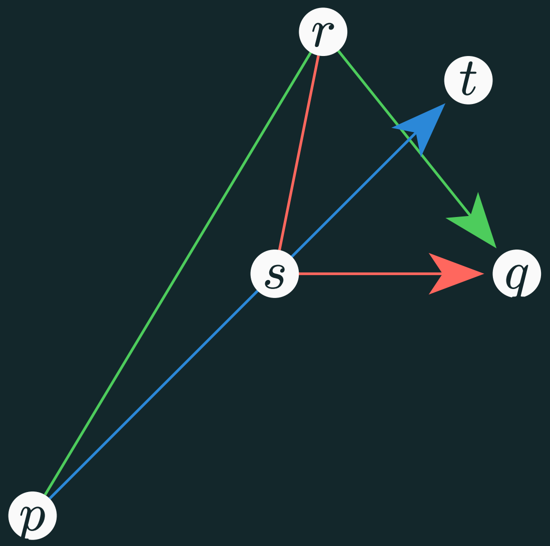
3. n points in canonical form with no 6-holes induce a propositional assignment that satisfies φ_n.

```
theorem satisfies_hexagonEncoding {w : CanonicalPoints} :
  ¬σHasEmptyKGon 6 w → w.toPropAssn ⊨ Geo.hexagonCNF w.len
```

4. But φ_{30} is unsatisfiable.

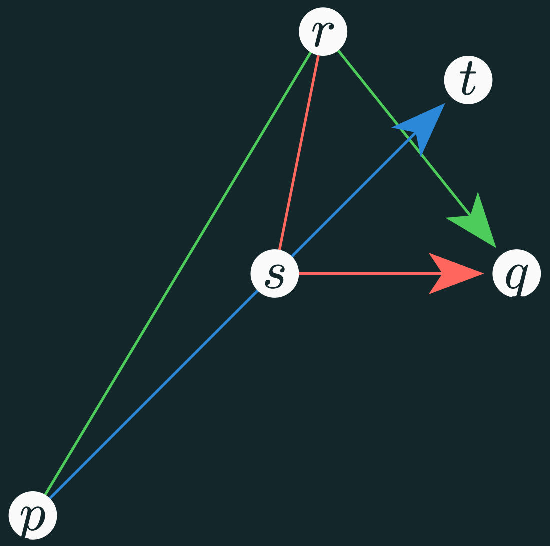
```
axiom unsat_6hole_cnf : (Geo.hexagonCNF 30).isUnsat
```

Discretization with triple-orientations



$$\sigma(p, r, q) = 1 \quad \text{(clockwise)}$$
$$\sigma(p, s, t) = 0 \quad \text{(collinear)}$$
$$\sigma(r, s, q) = -1 \quad \text{(counter-clockwise)}$$

Discretization with triple-orientations



$$\sigma(p, r, q) = 1 \quad \text{(clockwise)}$$
$$\sigma(p, s, t) = 0 \quad \text{(collinear)}$$
$$\sigma(r, s, q) = -1 \quad \text{(counter-clockwise)}$$

$\exists$ k-hole $\iff$ a propositional formula over $\sigma(a, b, c)$ is satisfiable

Reduction from geometry to SAT

1. Discretize from continuous coordinates in $\mathbb{R}^2$ to boolean variables.
2. Points can be put in *canonical form* without removing k-holes.

```
theorem symmetry_breaking {l : List Point} :
  3 ≤ l.length → PointsInGenPos l →
  ∃ w : CanonicalPoints, l ≼σ w.points
```

3. n points in canonical form with no 6-holes induce a propositional assignment that satisfies φ_n.

```
theorem satisfies_hexagonEncoding {w : CanonicalPoints} :
  ¬σHasEmptyKGon 6 w → w.toPropAssn ⊨ Geo.hexagonCNF w.len
```

4. But φ_{30} is unsatisfiable.

```
axiom unsat_6hole_cnf : (Geo.hexagonCNF 30).isUnsat
```

Symmetry breaking

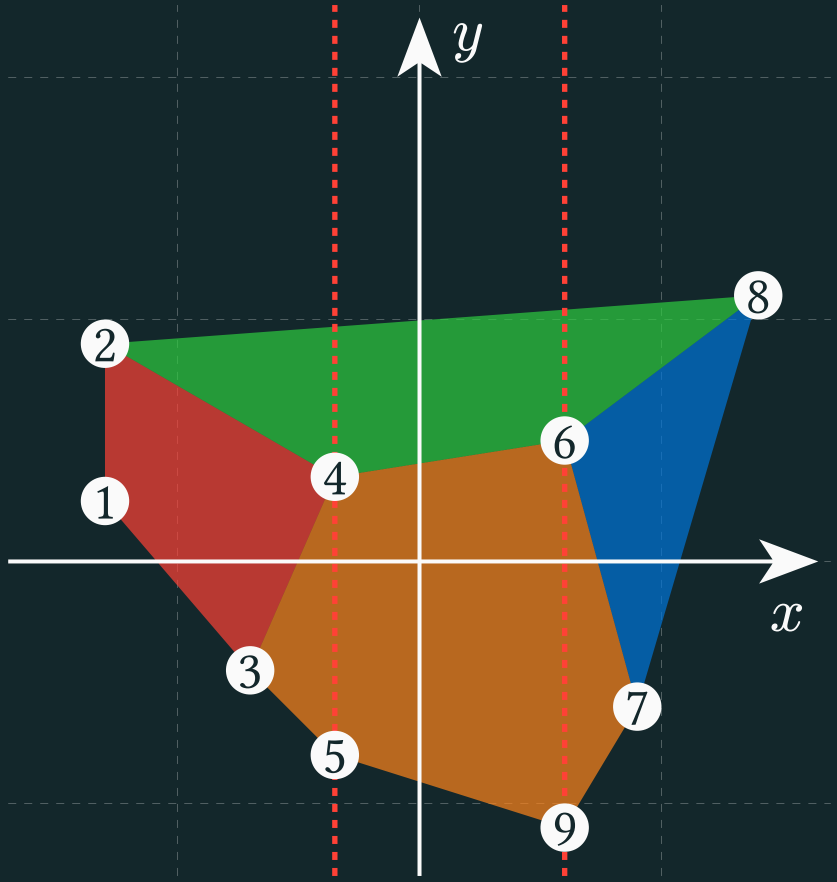
Lemma. WLOG we can assume that the points $(p_1, ..., p_n)$ are in *canonical form*, meaning that they satisfy the following properties:

- **(x-order)** The points are sorted with respect to their x-coordinates, i.e., $(p_i)_x < (p_j)_x$ for all $1 \leq i < j \leq n$.
- **(CCW-order)** All orientations $\sigma(p_1, p_i, p_j)$, with $1 < i < j \leq n$, are counterclockwise.
- **(Lex order)** The first half of list of adj. orientations is lex-below the second half:

$$\left[\sigma\left(p_{\lceil \frac{n}{2} \rceil+1}, p_{\lceil \frac{n}{2} \rceil+2}, p_{\lceil \frac{n}{2} \rceil+3}\right), ..., \sigma(p_{n-2}, p_{n-1}, p_n)\right] \succcurlyeq$$
$$\left[\sigma\left(p_{\lceil \frac{n}{2} \rceil-1}, p_{\lceil \frac{n}{2} \rceil}, p_{\lceil \frac{n}{2} \rceil+1}\right), ..., \sigma(p_2, p_3, p_4)\right]$$

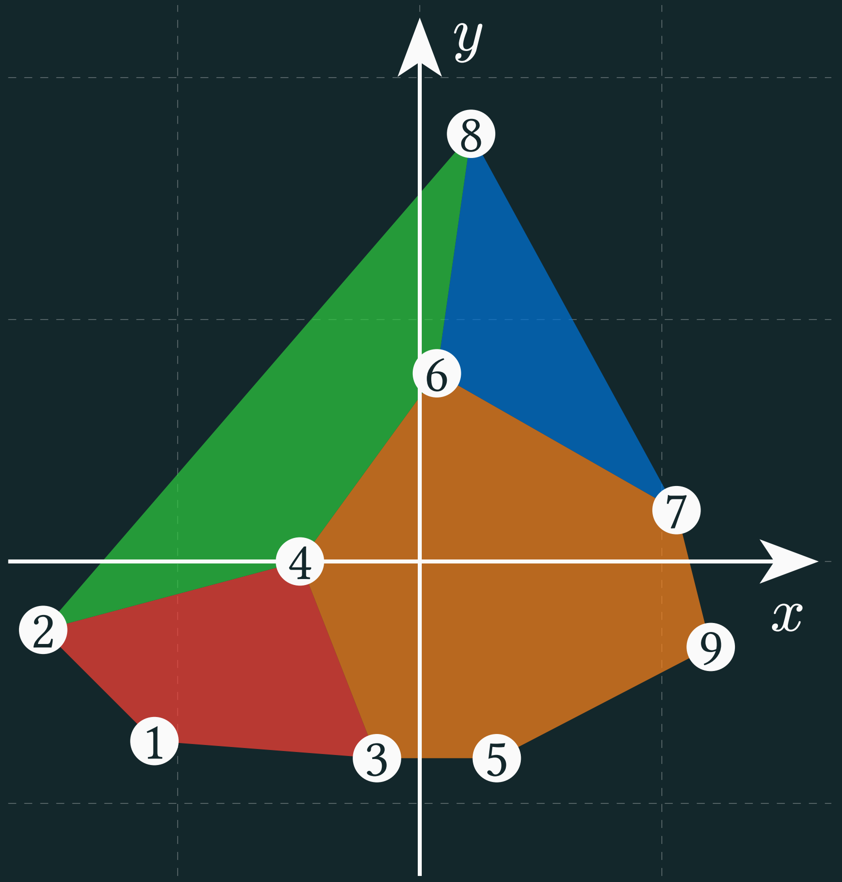
Symmetry breaking

Starting set of points.



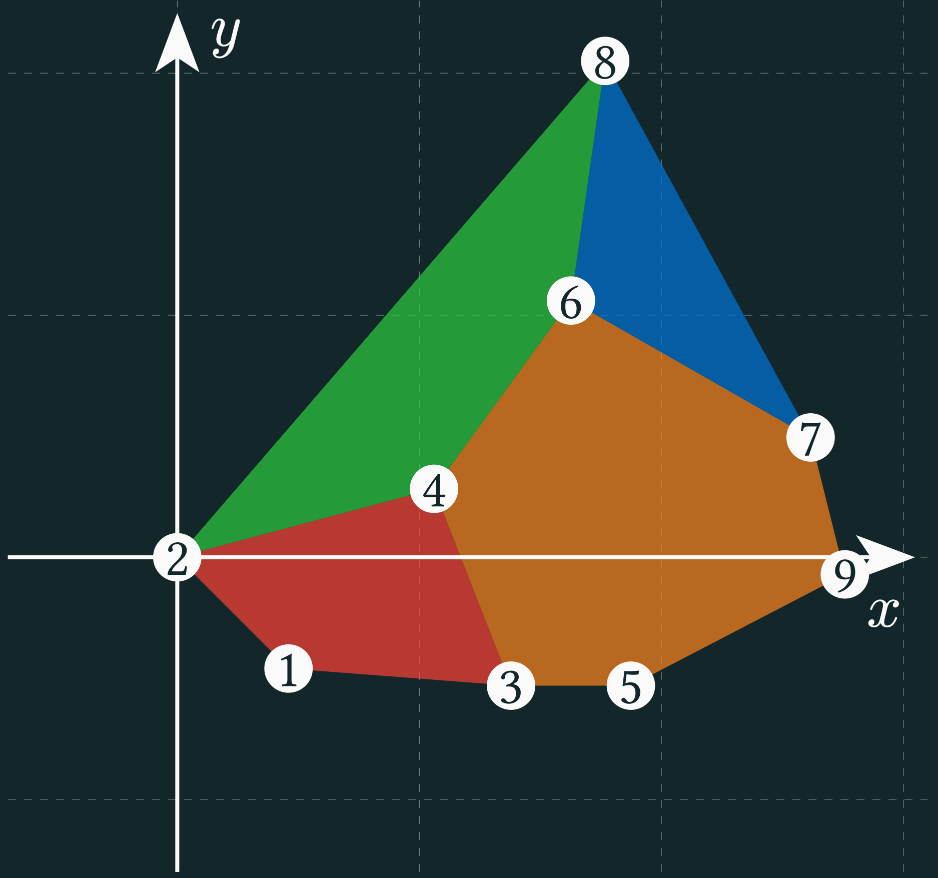
Symmetry breaking

Rotation ensures distinct x.



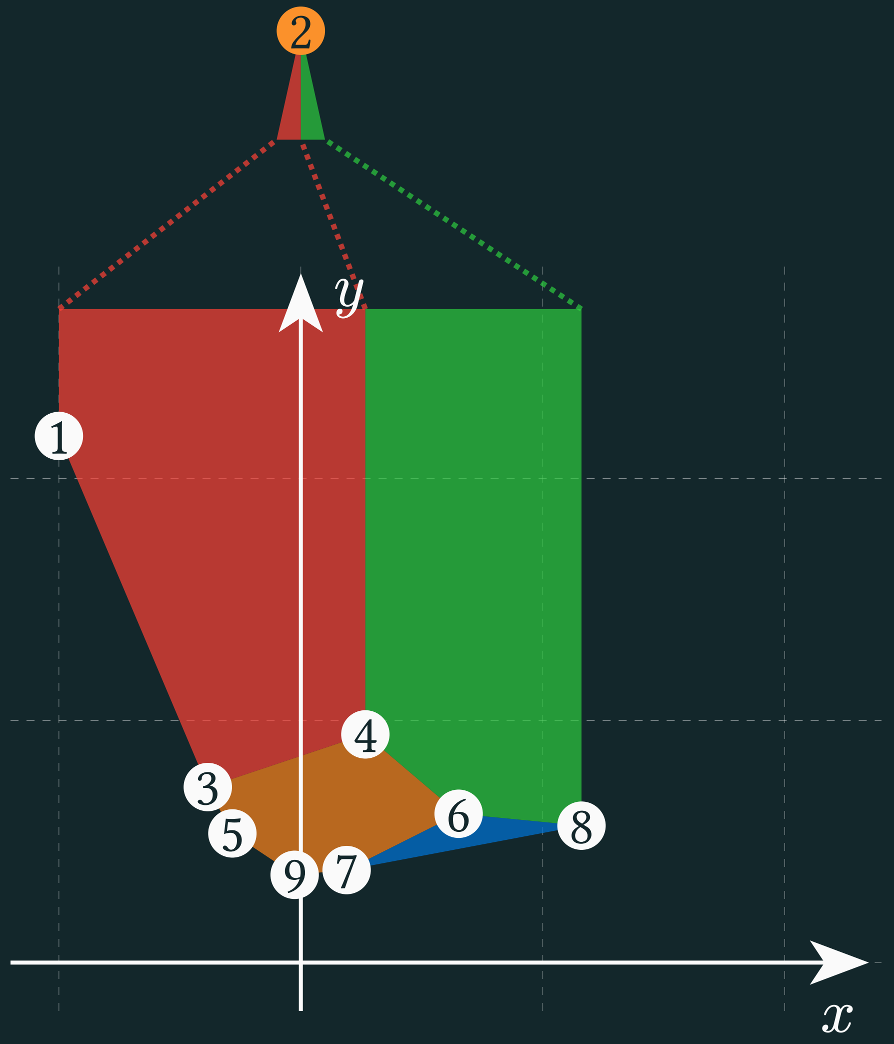
Symmetry breaking

Translate leftmost point to $(0, 0)$.
Ensures nonnegative x.



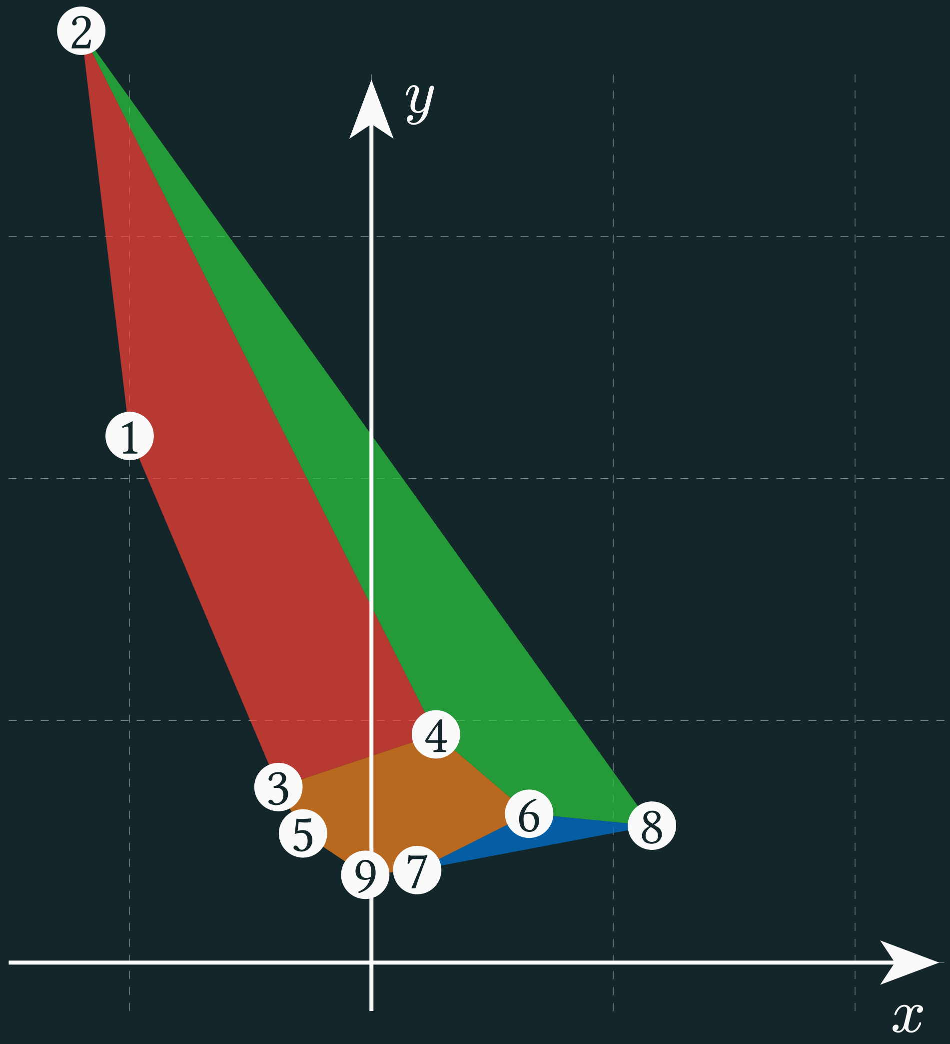
Symmetry breaking

Map $(x, y) \mapsto \left(\frac{y}{x}, \frac{1}{x}\right)$.



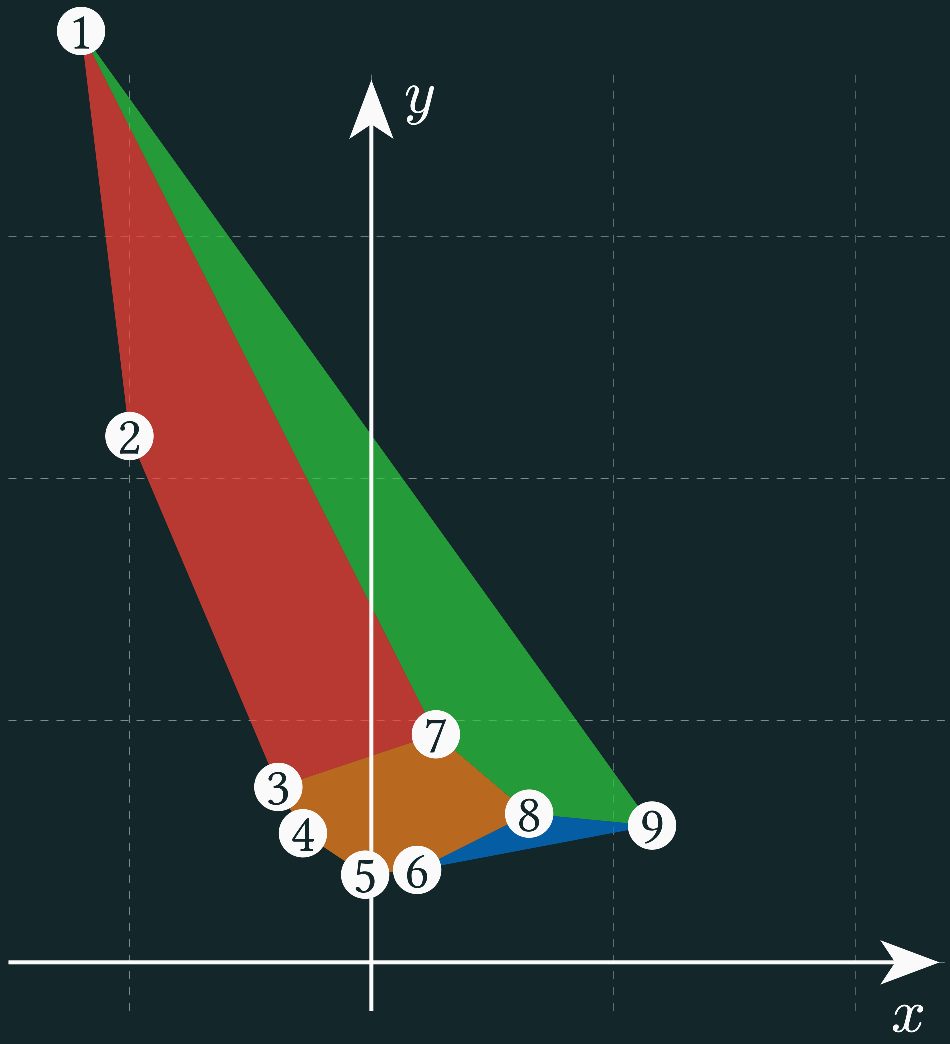
Symmetry breaking

Bring point at ∞ back.



Symmetry breaking

Relabel in order of increasing x.



Reduction from geometry to SAT

1. Discretize from continuous coordinates in $\mathbb{R}^2$ to boolean variables.
2. Points can be put in *canonical form* without removing *k*-holes.

```
theorem symmetry_breaking {l : List Point} :
  3 ≤ l.length → PointsInGenPos l →
  ∃ w : CanonicalPoints, l ≼σ w.points
```

3. n points in canonical form with no 6-holes induce a propositional assignment that satisfies φ_n.

```
theorem satisfies_hexagonEncoding {w : CanonicalPoints} :
  ¬σHasEmptyKGon 6 w → w.toPropAssn ⊨ Geo.hexagonCNF w.len
```

4. But φ_{30} is unsatisfiable.

```
axiom unsat_6hole_cnf : (Geo.hexagonCNF 30).isUnsat
```

Running the SAT solver

CNF formula produced directly from executable Lean definition.

Running the SAT solver

CNF formula produced directly from executable Lean definition.

To verify $h(6) \leq 30$:

Running the SAT solver

CNF formula produced directly from executable Lean definition.

To verify $h(6) \leq 30$:

- CNF with 65 092 variables and 436 523 clauses

Running the SAT solver

CNF formula produced directly from executable Lean definition.

To verify $h(6) \leq 30$:

- CNF with 65 092 variables and 436 523 clauses
- partitioned into 312 418 subproblems

Running the SAT solver

CNF formula produced directly from executable Lean definition.

To verify $h(6) \leq 30$:

- CNF with 65 092 variables and 436 523 clauses
- partitioned into 312 418 subproblems
- each subproblem solved by CaDiCaL 1.9.5

Running the SAT solver

CNF formula produced directly from executable Lean definition.

To verify $h(6) \leq 30$:

- CNF with 65 092 variables and 436 523 clauses
- partitioned into 312 418 subproblems
- each subproblem solved by CaDiCaL 1.9.5
- unsatisfiability proofs checked on-the-fly by `cake_lpr`

Running the SAT solver

CNF formula produced directly from executable Lean definition.

To verify $h(6) \leq 30$:

- CNF with 65 092 variables and 436 523 clauses
- partitioned into 312 418 subproblems
- each subproblem solved by CaDiCaL 1.9.5
- unsatisfiability proofs checked on-the-fly by `cake_lpr`
- 25 876.5 CPU hours on Bridges 2 cluster of Pittsburgh Supercomputing Center

Lower bounds

To prove $n < h(k)$, find a set of n points with no k-holes.

Lower bounds

To prove $n < h(k)$, find a set of n points with no k-holes.

Naive checker algorithm is $\mathcal{O}\left(n^{k+1} \log k\right)$ time.

Lower bounds

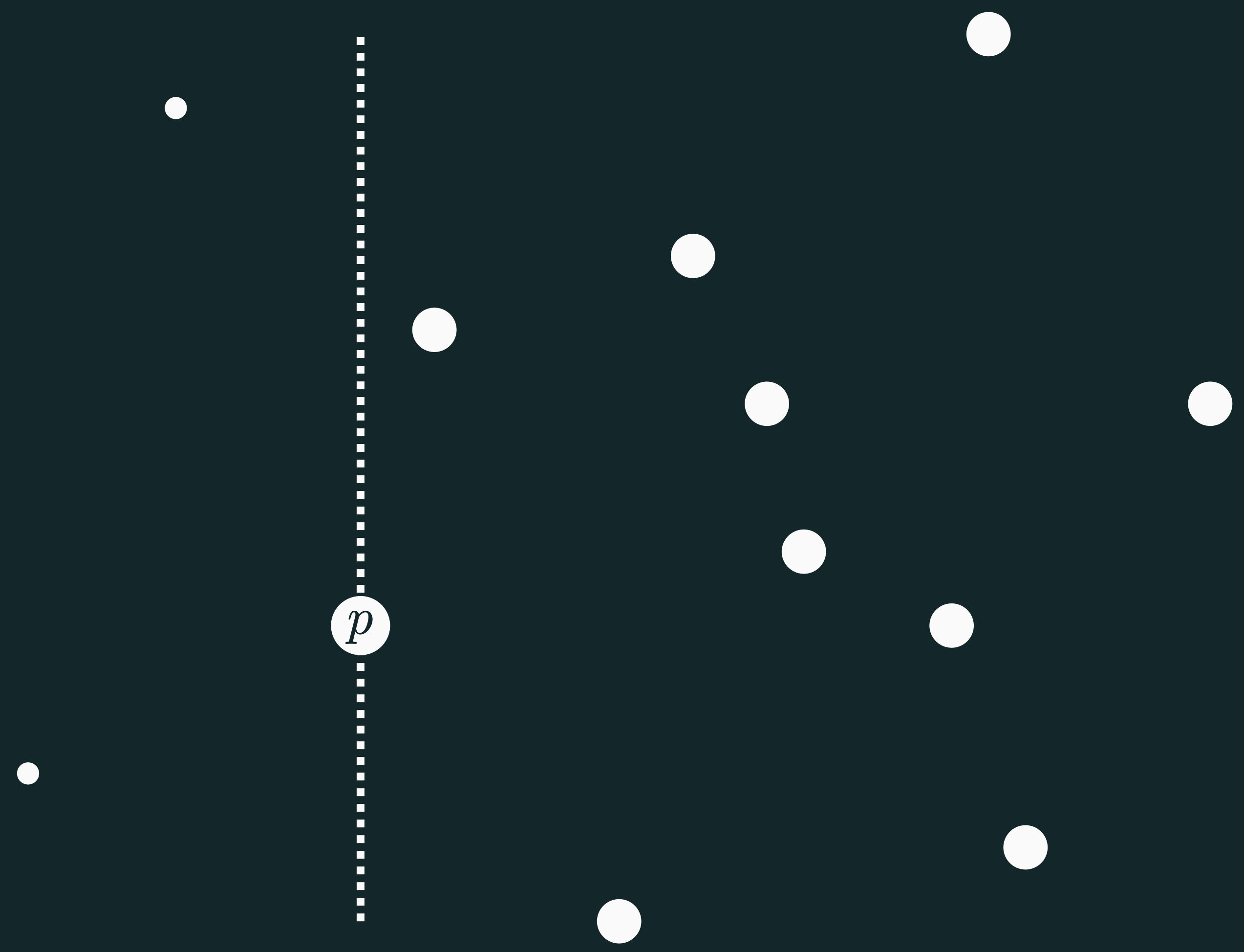
To prove $n < h(k)$, find a set of n points with no k-holes.

Naive checker algorithm is $\mathcal{O}\left(n^{k+1} \log k\right)$ time.

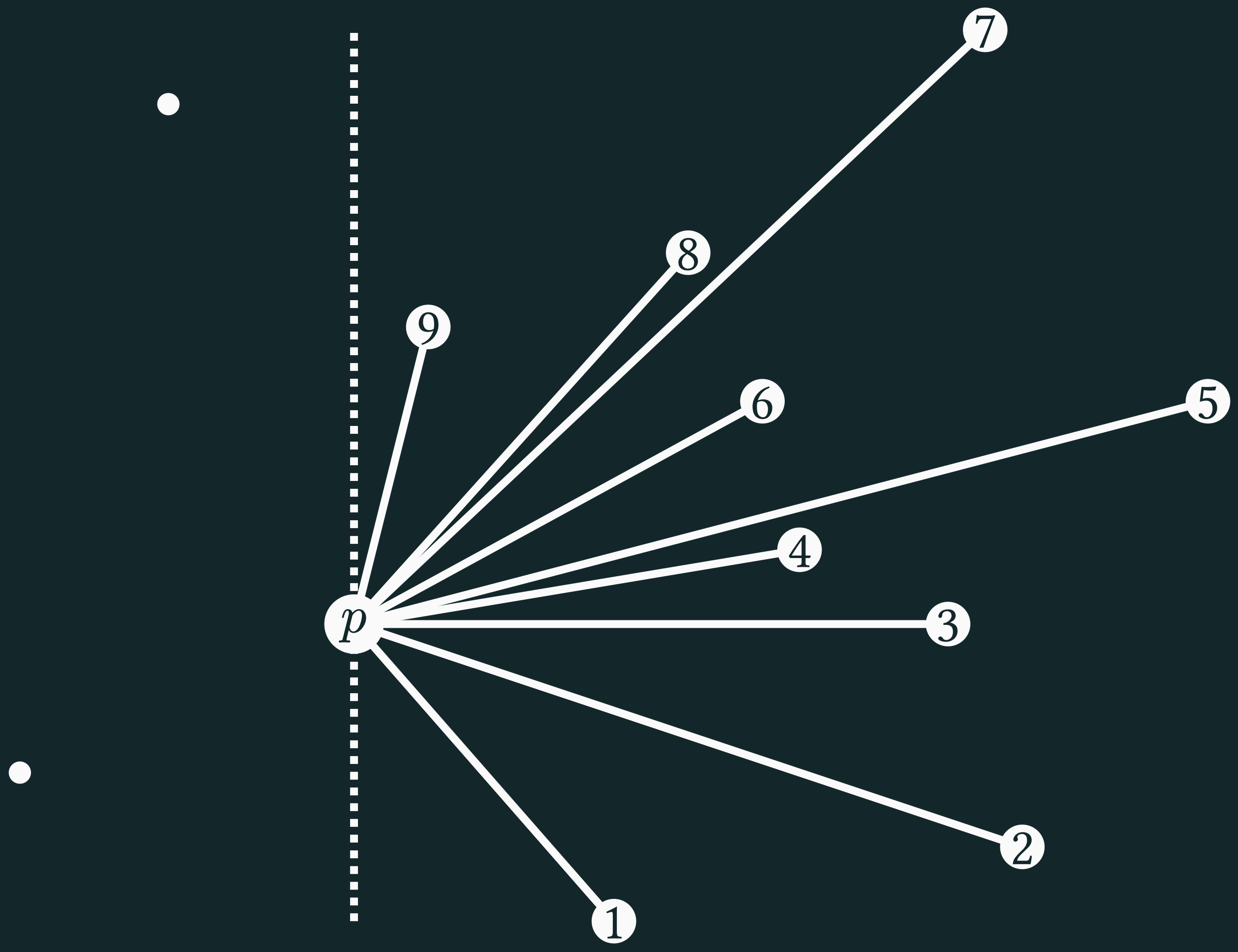
We verified an $\mathcal{O}(n^3)$ solution
from Dobkin, Edelsbrunner, and Overmars (1990).

Hole-finding algorithm

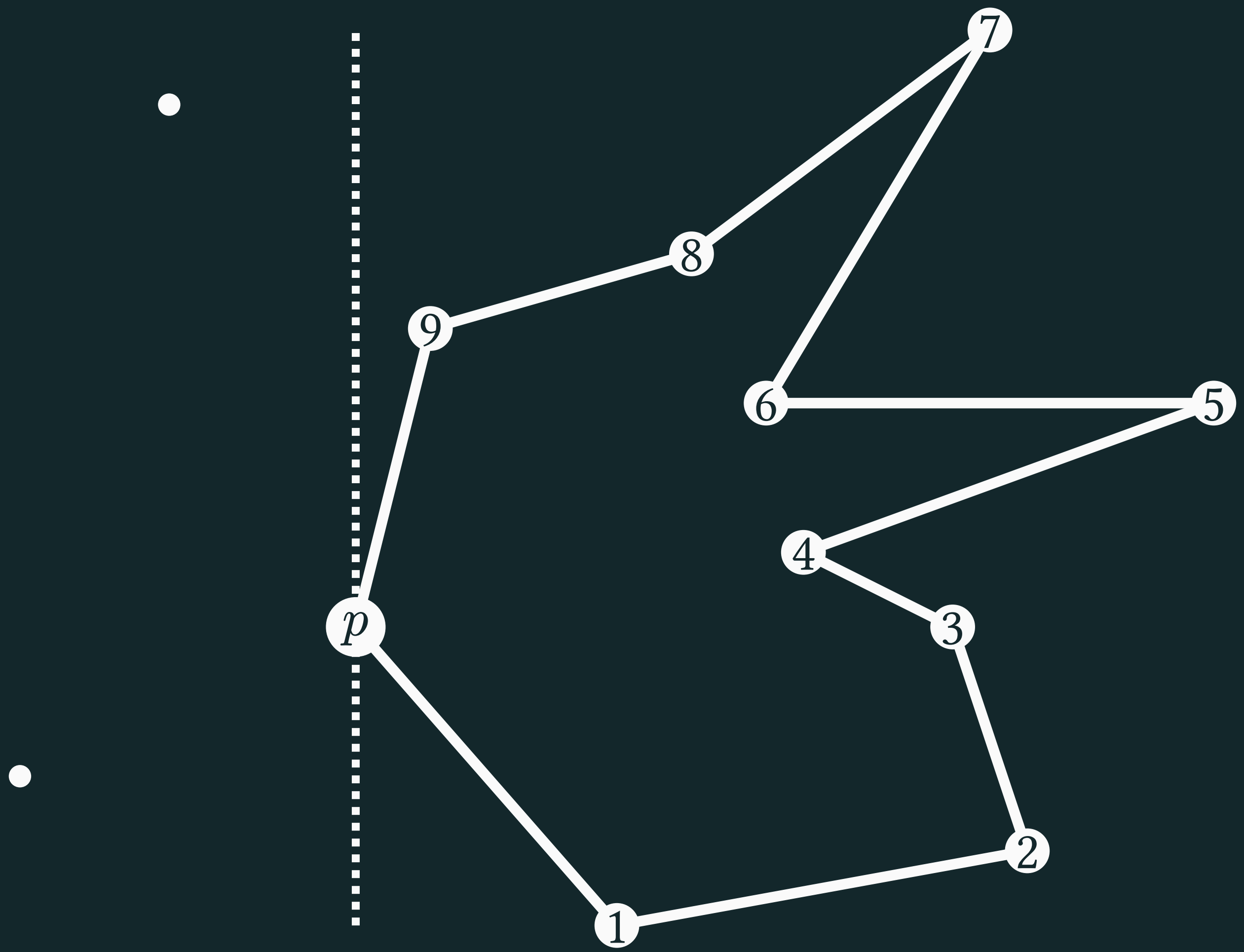
Hole-finding algorithm



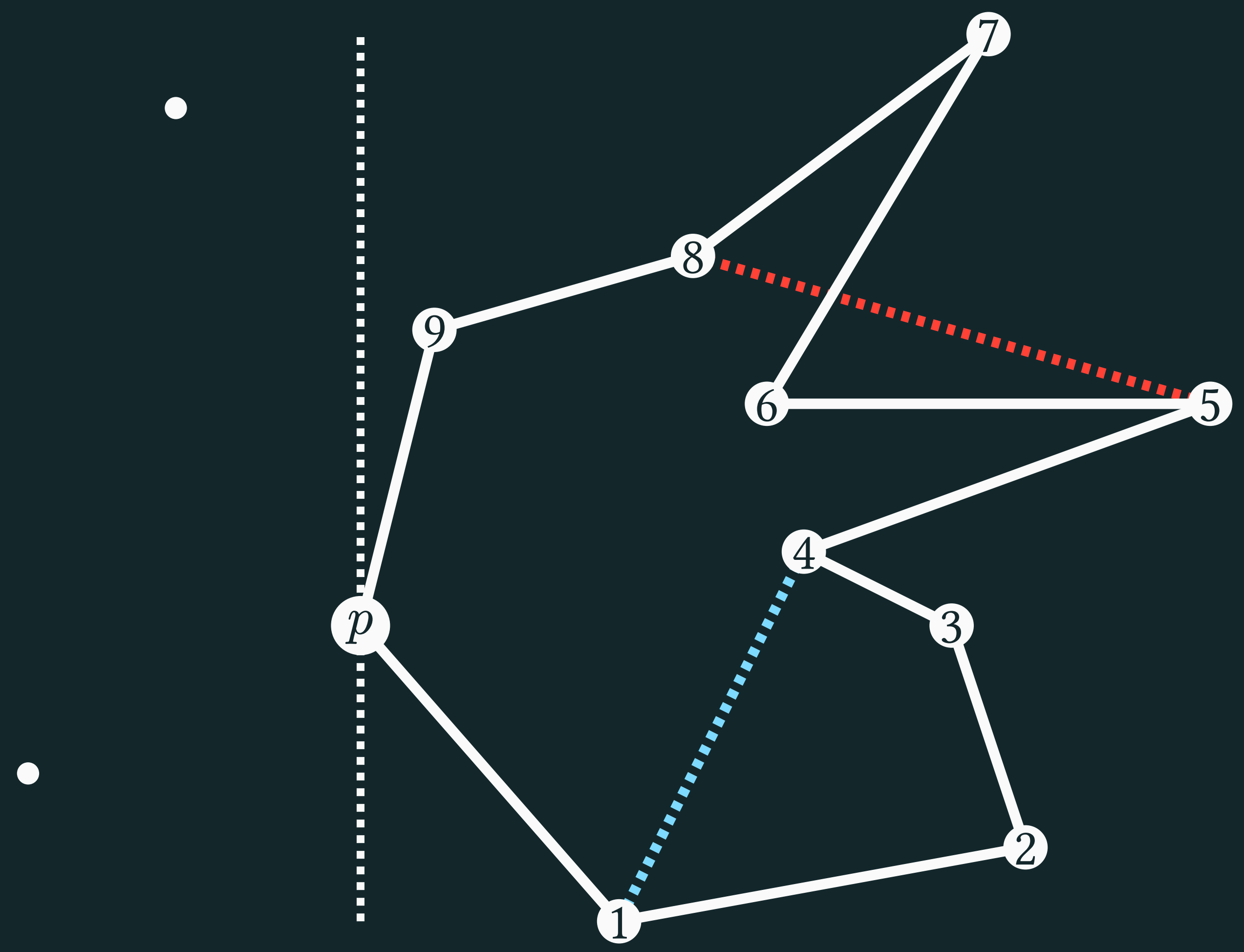
Hole-finding algorithm



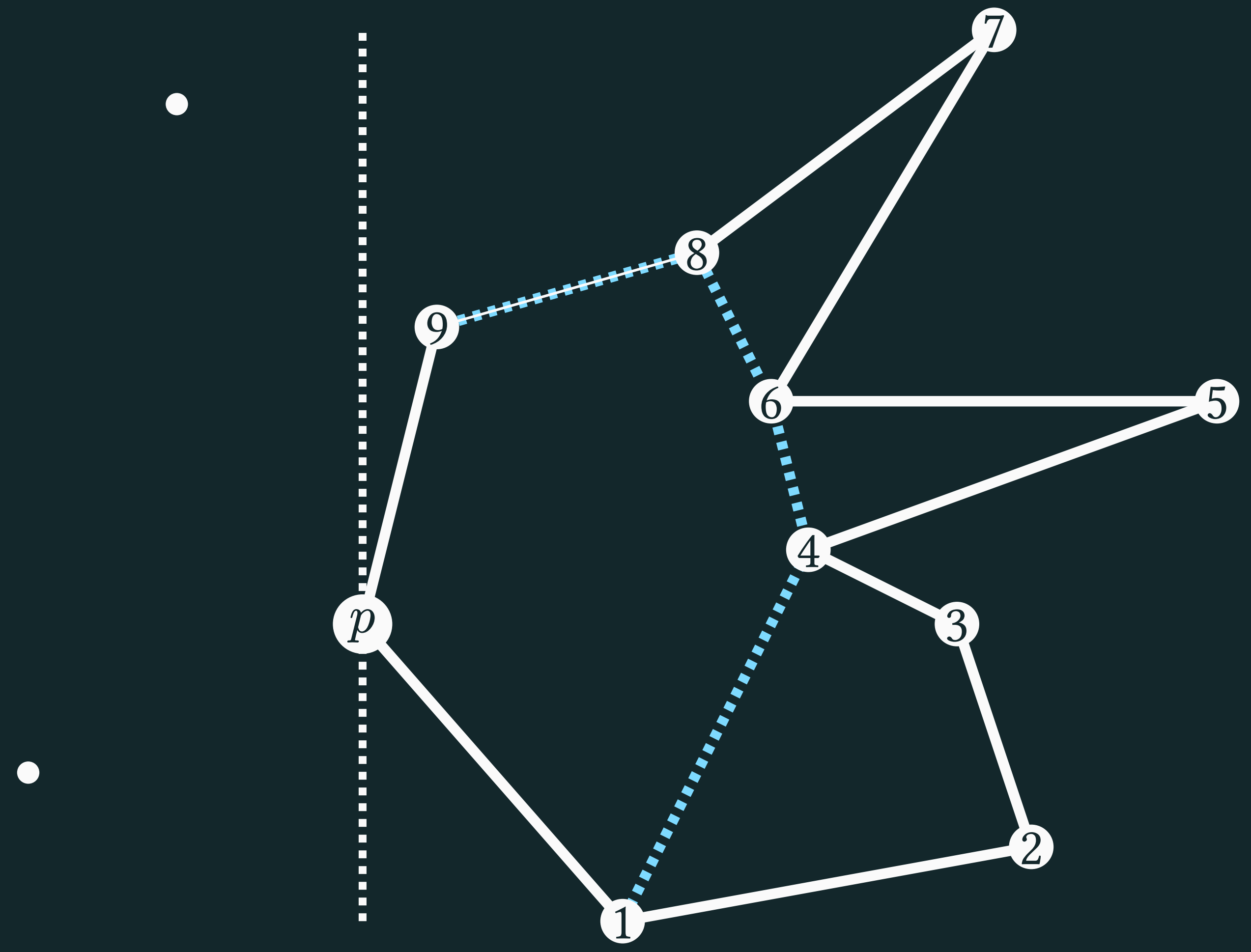
Hole-finding algorithm



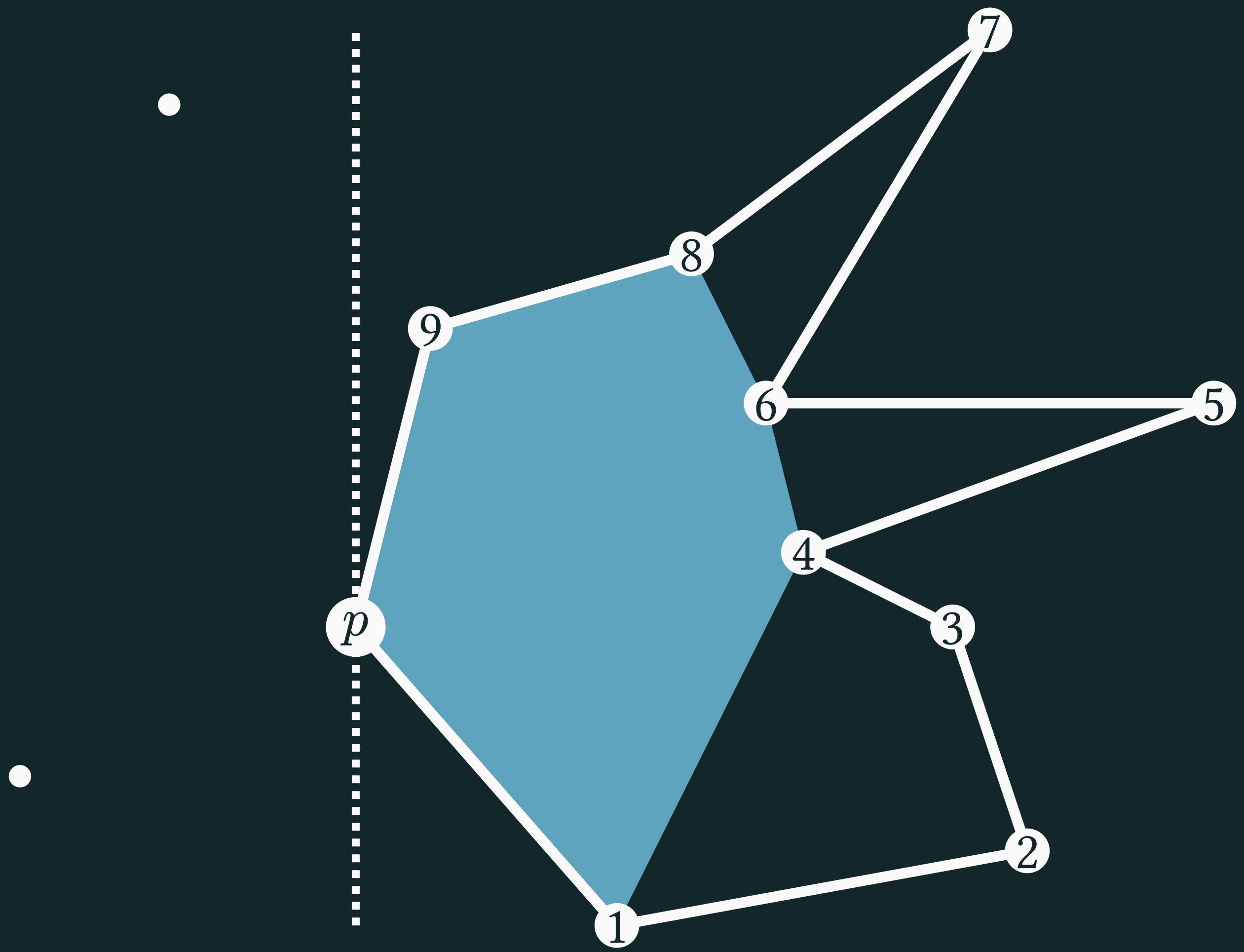
Hole-finding algorithm



Hole-finding algorithm



Hole-finding algorithm



Hole-finding algorithm: verification

If $j = i + 1$ $\overline{ij}$ clearly is in VG.

Hole-finding algorithm: verification

If $j = i + 1$ $\overline{ij}$ clearly is in VG.

expands to

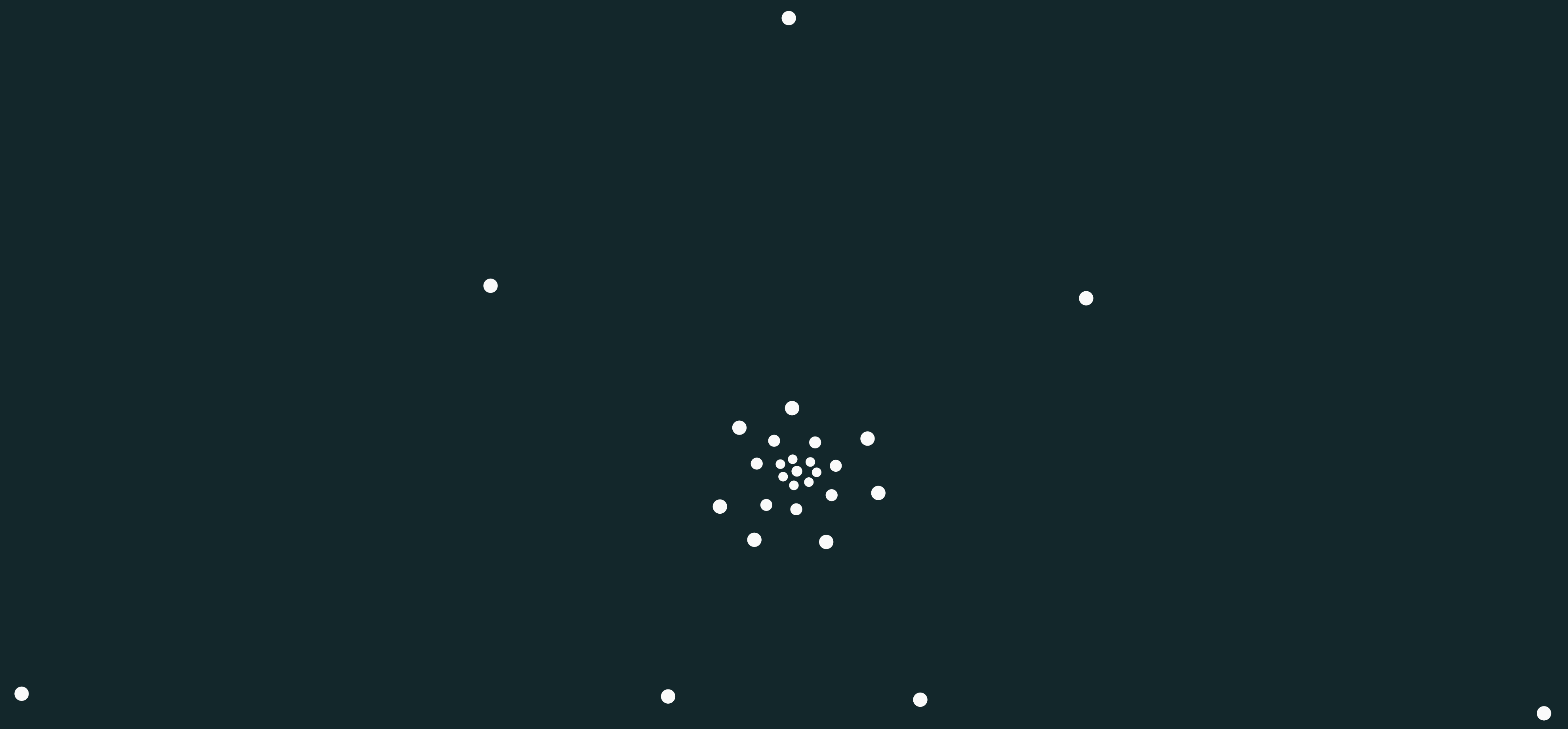
Hole-finding algorithm: verification

If $j = i + 1$ $\overline{ij}$ clearly is in VG.

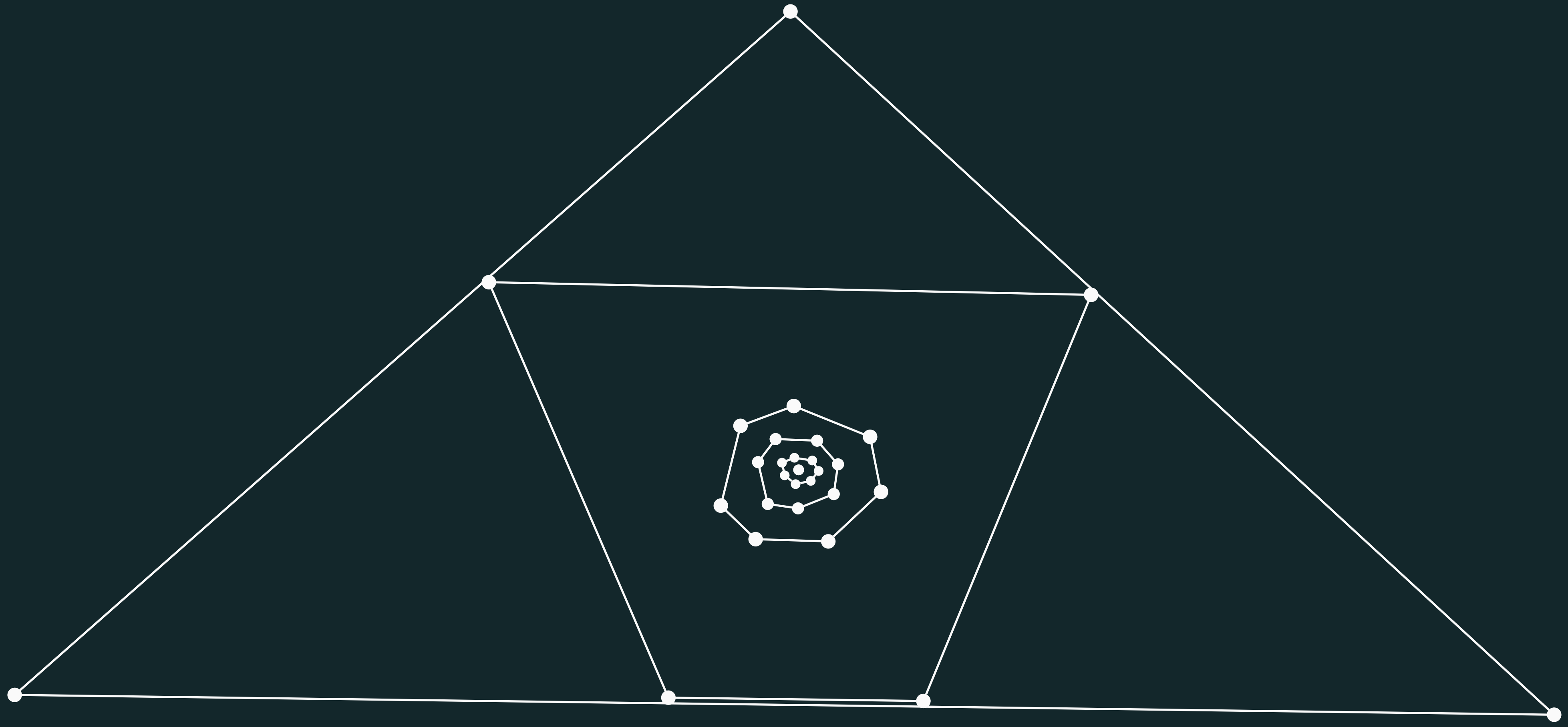
expands to

```
theorem of_proceed_loop
    {i j : Fin n} (ij : Visible p pts i j) {Q : Queues n j} {Q_j : BelowList n j} {Q_i} (ha)
    {IH} (hIH : ∀ a (ha : a < i), Visible p pts a j → ProceedIH p pts (ha.trans ij.1) (IH a ha))
    (Hj : Queues.OrderedTail p pts lo j (fun k h => Q.q[k.1]'(Q.sz ▸ h)) Q_j.1)
    (ord : Queues.Ordered p pts lo i (fun k h => Q.q[k.1]'(Q.sz ▸ h.trans ij.1)) Q_i)
    (g_wf : Q.graph.WF (VisibleLT p pts lo j))
    {Q' Q_j'} (eq : proceed.loop pts i j ij.1 IH Q Q_j Q_i ha = (Q', Q_j')) :
    ∃ a Q₁ Q_i₁ Q_j₁, proceed.finish i j ij.1 Q₁ Q_i₁ Q_j₁ = (Q', Q_j') ∧
      Q₁.graph.WF (VisibleLT p pts i j) ∧
      (∀ k ∈ Q_i₁.1, σ (pts k) (pts i) (pts j) ≠ .ccw) ∧
      lo ≤ a ∧ Queues.Ordered p pts a i (fun k h => Q.q[k.1]'(Q.sz ▸ h.trans ij.1)) Q_i₁.1 ∧
      (∀ (k : Fin n) (h : k < j), ¬(lo ≤ k ∧ k < a) → Q₁.q[k.1]'(Q₁.sz ▸ h) = Q.q[k.1]'(Q.sz ▸ h)) ∧
      Queues.OrderedTail p pts a j (fun k h => Q₁.q[k.1]'(Q₁.sz ▸ h)) Q_j₁.1 := by
```

Lower bound: 29 points with no 6-holes (Overmars 2003)



Lower bound: 29 points with no 6-holes (Overmars 2003)



Final theorem

```
axiom unsat_6hole_cnf : (Geo.hexagonCNF 30).isUnsat

theorem holeNumber_6 : holeNumber 6 = 30 :=
  le_antisymm
   (hole_6_theorem' unsat_6hole_cnf)
   (hole_lower_bound [
    (1, 1260),  (16, 743),  (22, 531),  (37, 0),    (306, 592),
    (310, 531), (366, 552), (371, 487), (374, 525), (392, 575),
    (396, 613), (410, 539), (416, 550), (426, 526), (434, 552),
    (436, 535), (446, 565), (449, 518), (450, 498), (453, 542),
    (458, 526), (489, 537), (492, 502), (496, 579), (516, 467),
    (552, 502), (754, 697), (777, 194), (1259, 320)])
```

Conclusion

We used Lean to fully verify a recent result in combinatorial geometry based on a sophisticated reduction to SAT.

Conclusion

We used Lean to fully verify a recent result in combinatorial geometry based on a sophisticated reduction to SAT.

Upper and lower bounds for all finite *hole numbers* $h(k)$ followed with additional effort.

Conclusion

We used Lean to fully verify a recent result in combinatorial geometry based on a sophisticated reduction to SAT.

Upper and lower bounds for all finite *hole numbers* $h(k)$ followed with additional effort.

Open problems remain:

- Horton's construction of $h(k) = \infty$ for $7 \leq k$ hasn't been verified.

Conclusion

We used Lean to fully verify a recent result in combinatorial geometry based on a sophisticated reduction to SAT.

Upper and lower bounds for all finite *hole numbers* $h(k)$ followed with additional effort.

Open problems remain:

- Horton's construction of $h(k) = \infty$ for $7 \leq k$ hasn't been verified.
- Exact values of $g(k) < \infty$ for $7 \leq k$ aren't known.

Conclusion

We used Lean to fully verify a recent result in combinatorial geometry based on a sophisticated reduction to SAT.

Upper and lower bounds for all finite *hole numbers* $h(k)$ followed with additional effort.

Open problems remain:

- Horton's construction of $h(k) = \infty$ for $7 \leq k$ hasn't been verified.
- Exact values of $g(k) < \infty$ for $7 \leq k$ aren't known.
- Trust story for large SAT proofs could be improved.

Bibliography

Dobkin, David P., Herbert Edelsbrunner, and Mark H. Overmars. 1990. "Searching for Empty Convex Polygons". *Algorithmica* 5 (4): 561–71. https://doi.org/10.1007/BF01840404

Erdős, Paul, and György Szekeres. 1935. "A Combinatorial Problem in Geometry". *Compositio Mathematica* 2: 463–70. http://eudml.org/doc/88611

Harborth, Heiko. 1978. "Konvexe Fünfecke in ebenen Punktmengen.". *Elemente Der Mathematik* 33: 116–18. http://eudml.org/doc/141217

Heule, Marijn J. H., and Manfred Scheucher. 2024. "Happy Ending: An Empty Hexagon in Every Set of 30 Points". In *Tools and Algorithms for the Construction and Analysis of Systems - 30th International Conference, TACAS 2024, Held as Part of the European Joint Conferences on Theory and Practice of*

Software, ETAPS 2024, Luxembourg City, Luxembourg, April 6-11, 2024, Proceedings, Part I, edited by Bernd Finkbeiner and Laura Kovács, 14570:61–80. Lecture Notes in Computer Science. Springer. https://doi.org/10.1007/978-3-031-57246-3_5

Horton, J. D. 1983. “Sets with No Empty Convex 7-Gons”. *Canadian Mathematical Bulletin* 26 (4): 482–84. https://doi.org/10.4153/CMB-1983-077-8

Marić, Filip. 2019. “Fast Formal Proof of the Erdős-Szekeres Conjecture for Convex Polygons with at Most 6 Points”. *J. Autom. Reason.* 62 (3): 301–29. https://doi.org/10.1007/S10817-017-9423-7

Bibliography

Overmars, Mark H. 2003. "Finding Sets of Points Without Empty Convex 6-Gons". *Discret. Comput. Geom.* 29 (1): 153–58. https://doi.org/10.1007/S00454-002-2829-X

Szekeres, George, and Lindsay Peters. 2006. "Computer Solution to the 17-Point Erdős-Szekeres Problem". *The ANZIAM Journal* 48 (2): 151–64. https://doi.org/10.1017/S144618110000300X