Edgebreaker

Connectivity Compression for Triangle Meshes

Jarek Rossignac, TVCG 1999

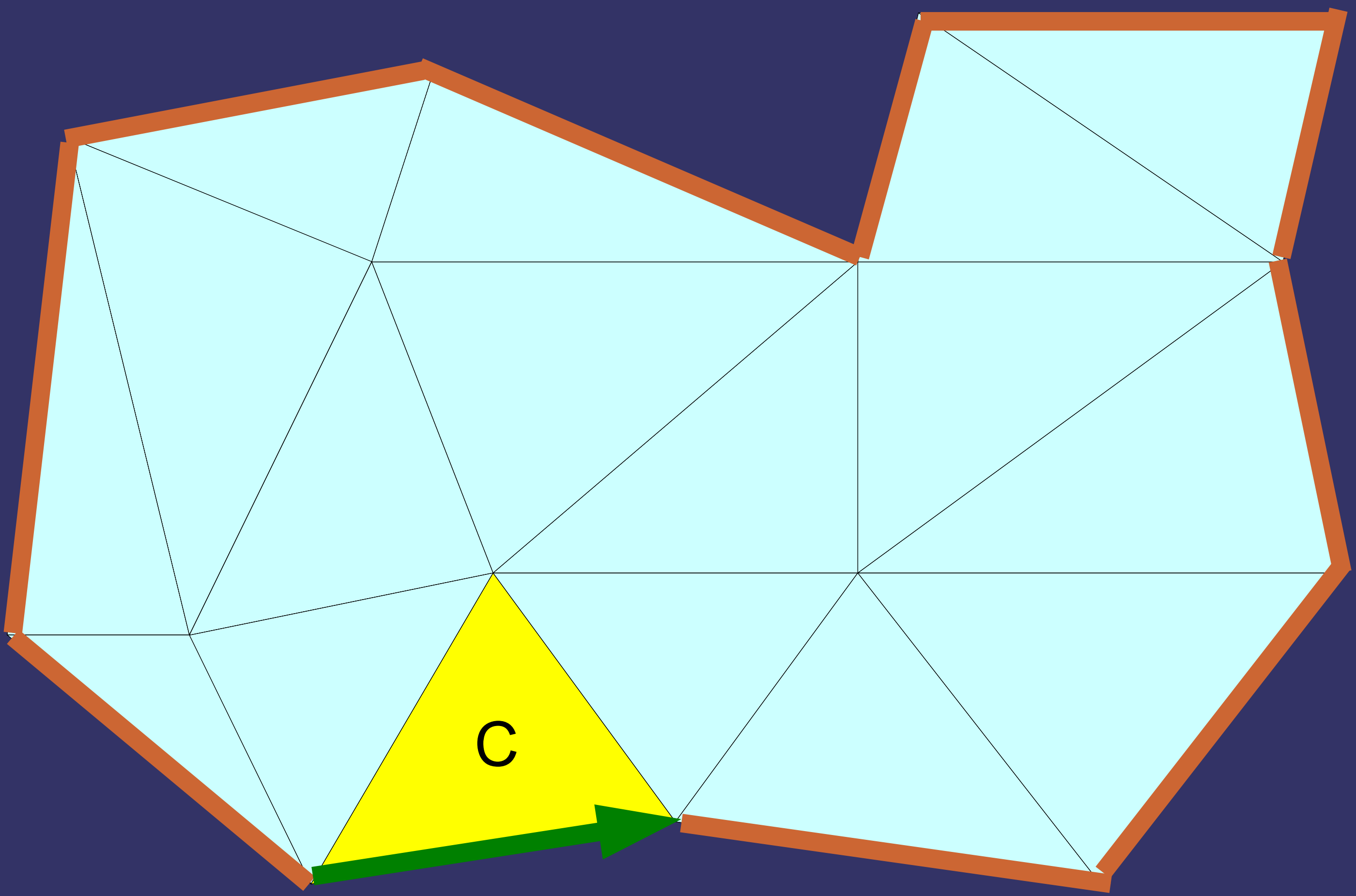
Contribution

- Lossless compression for a triangle mesh, using ≈2 bits per triangle for simple meshes
  - Only a slight increase for meshes with holes and handles
- Linear encoding size $O(|T|)$
  - Improves upon $O(|T| \log |T|)$ for many previous approaches
  - The constant is better than for previous approaches

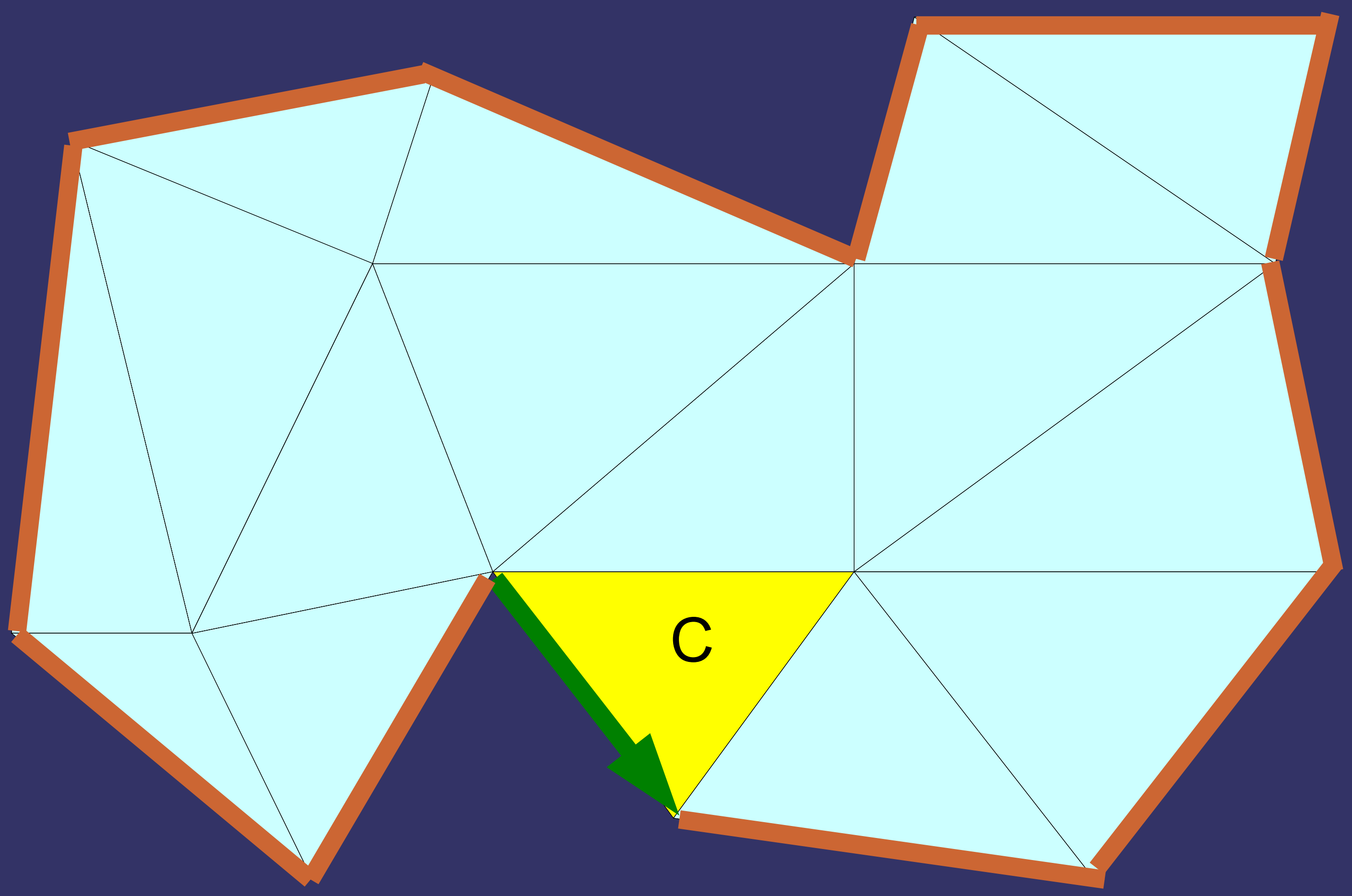
Basic Idea

- Destroy triangles of the mesh one-by-one, starting from the boundary and spiralling inwards
- For each destruction operation, store an opcode indicating the type of the operation
  - Sequence of opcodes is called “history”
  - Length of history == number of triangles, hence linear size encoding

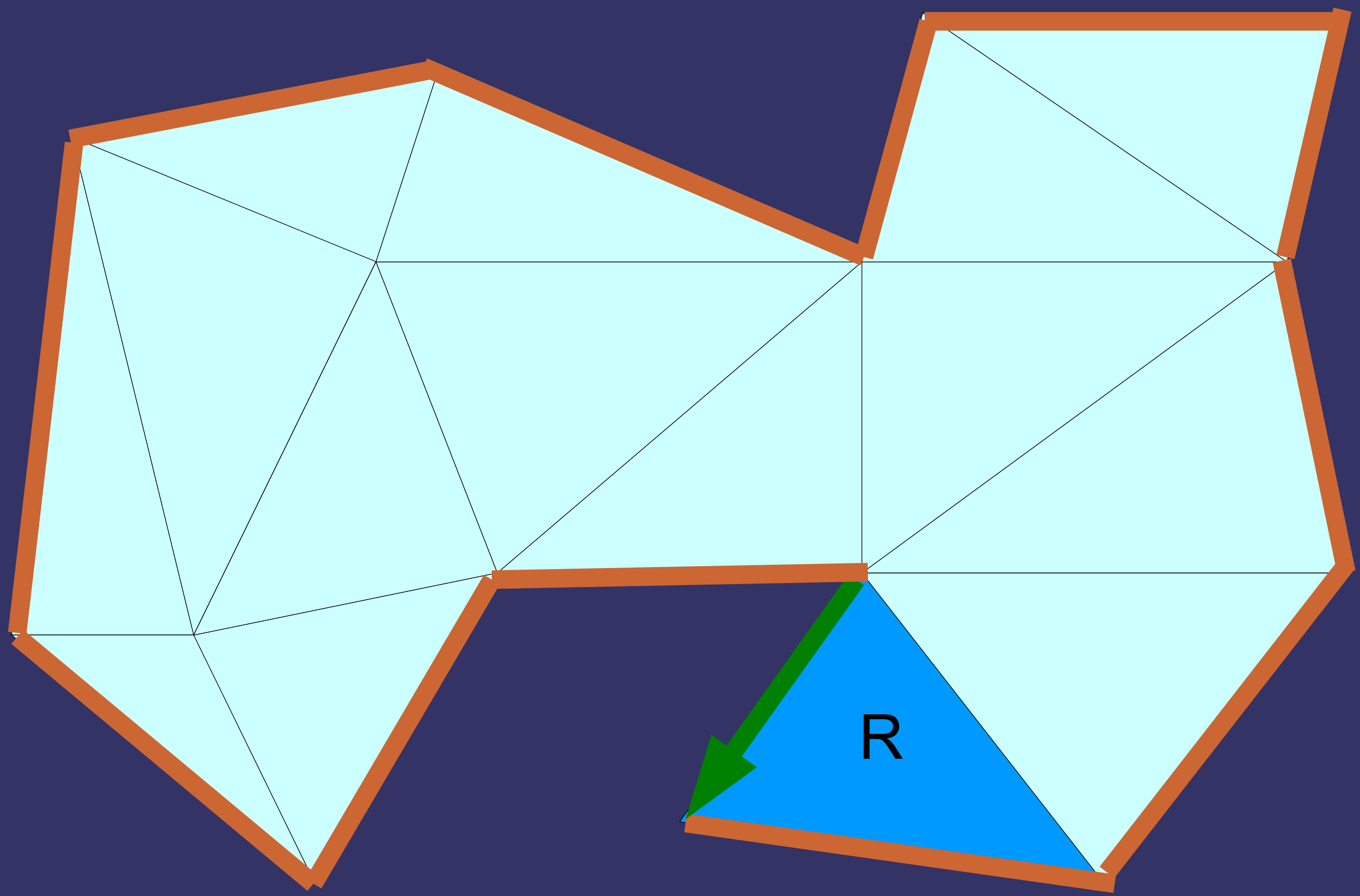
Edgebreaker in action



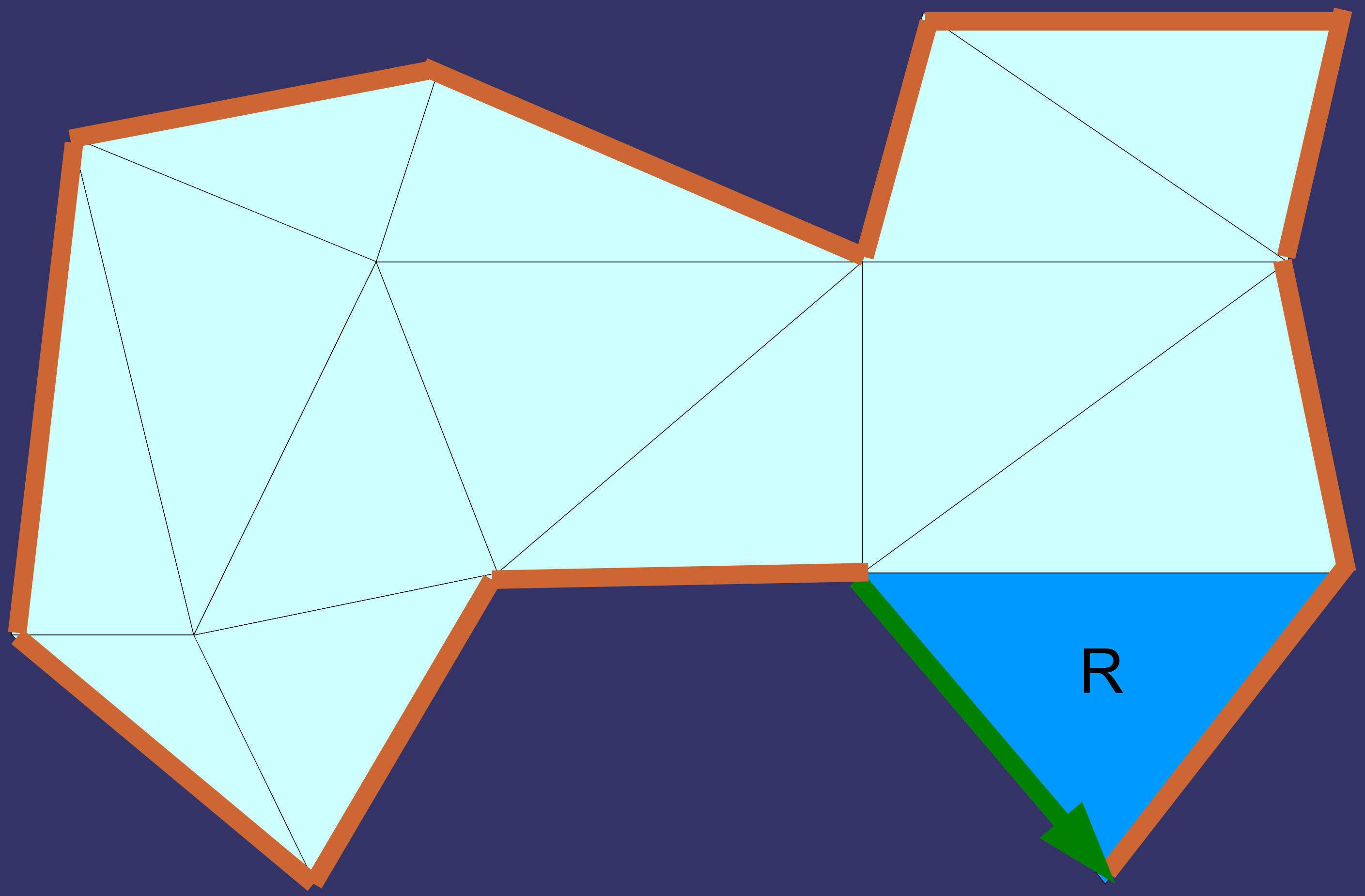
Edgebreaker in action



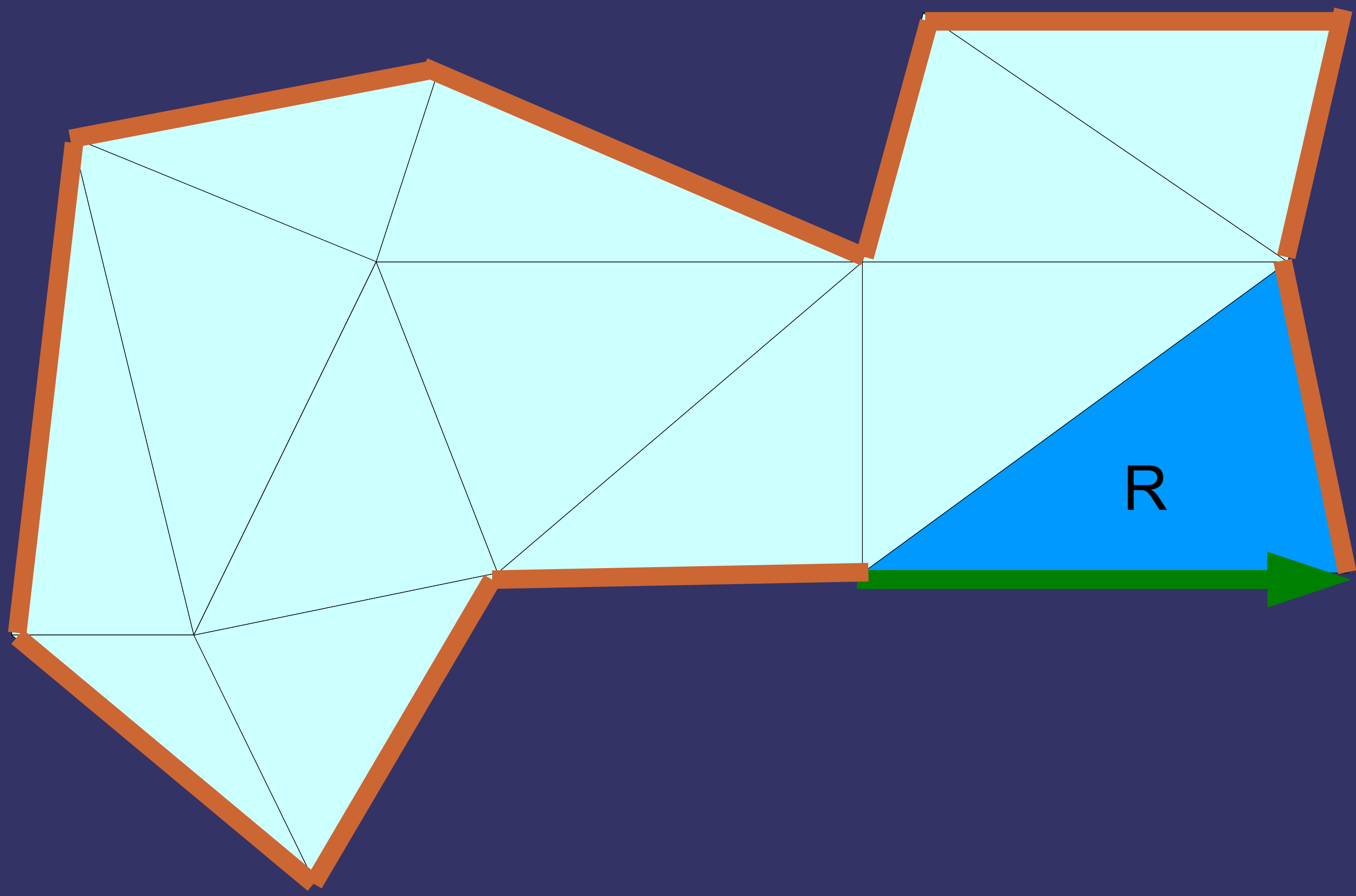
Edgebreaker in action



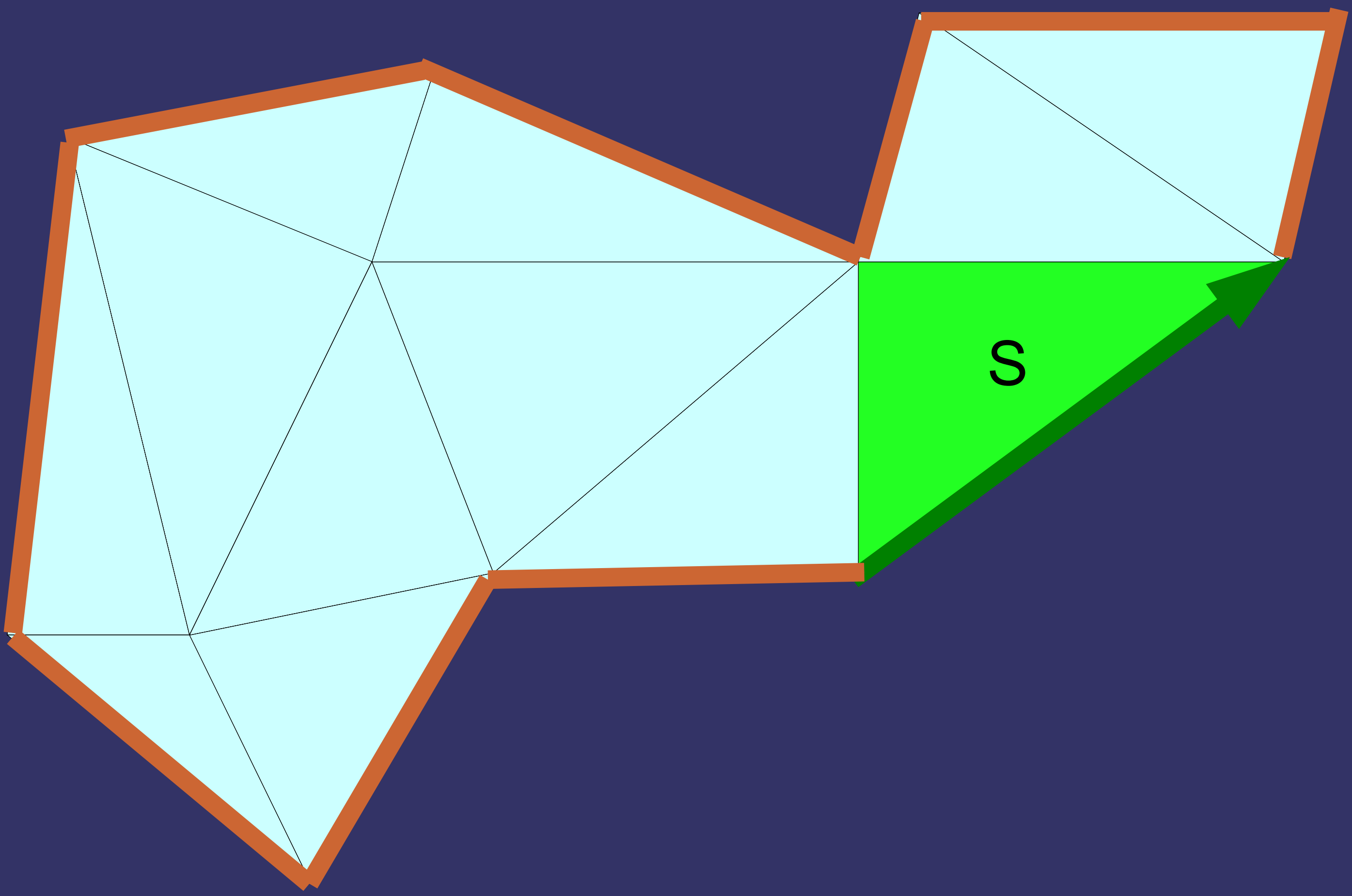
Edgebreaker in action



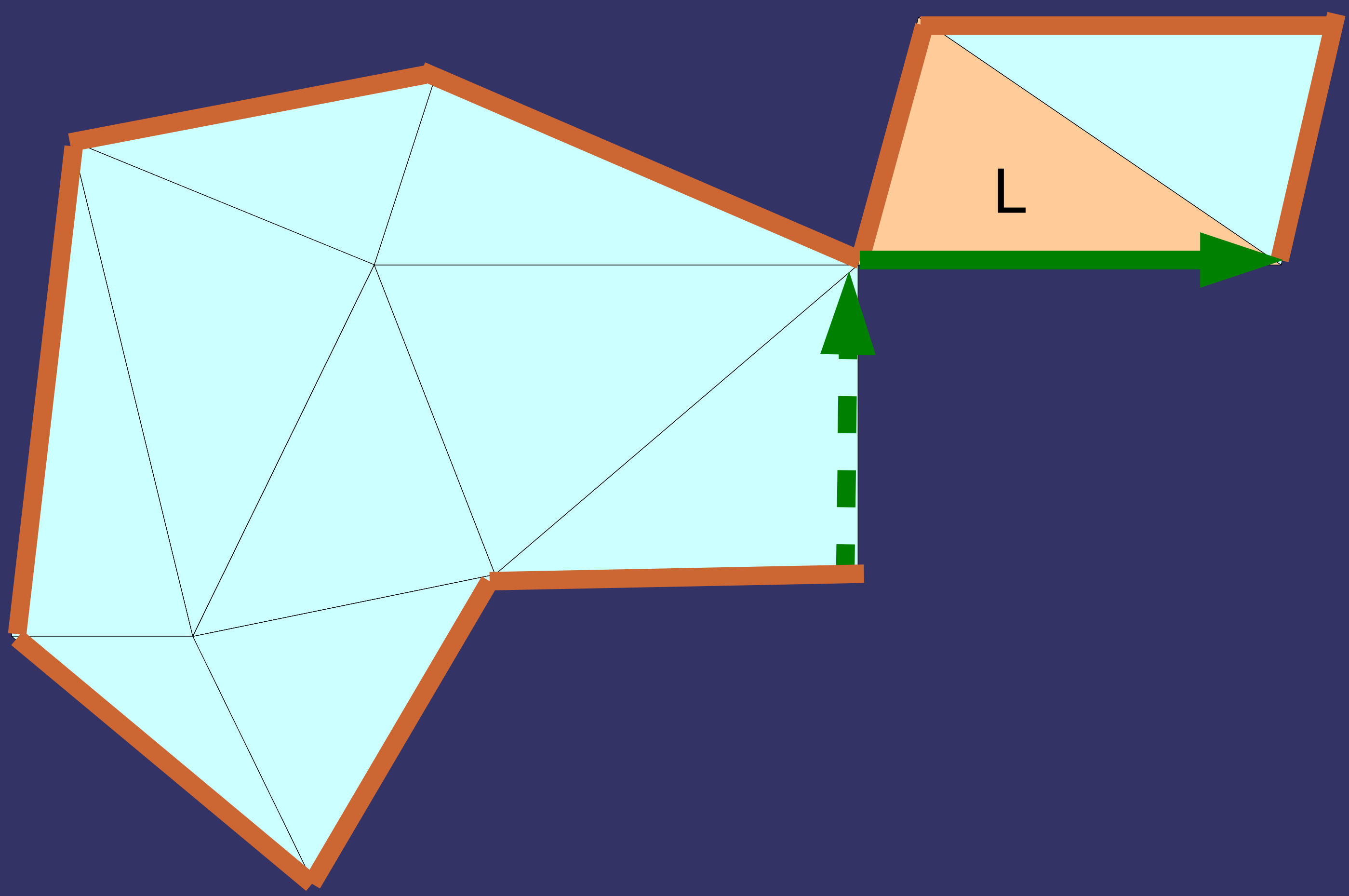
Edgebreaker in action



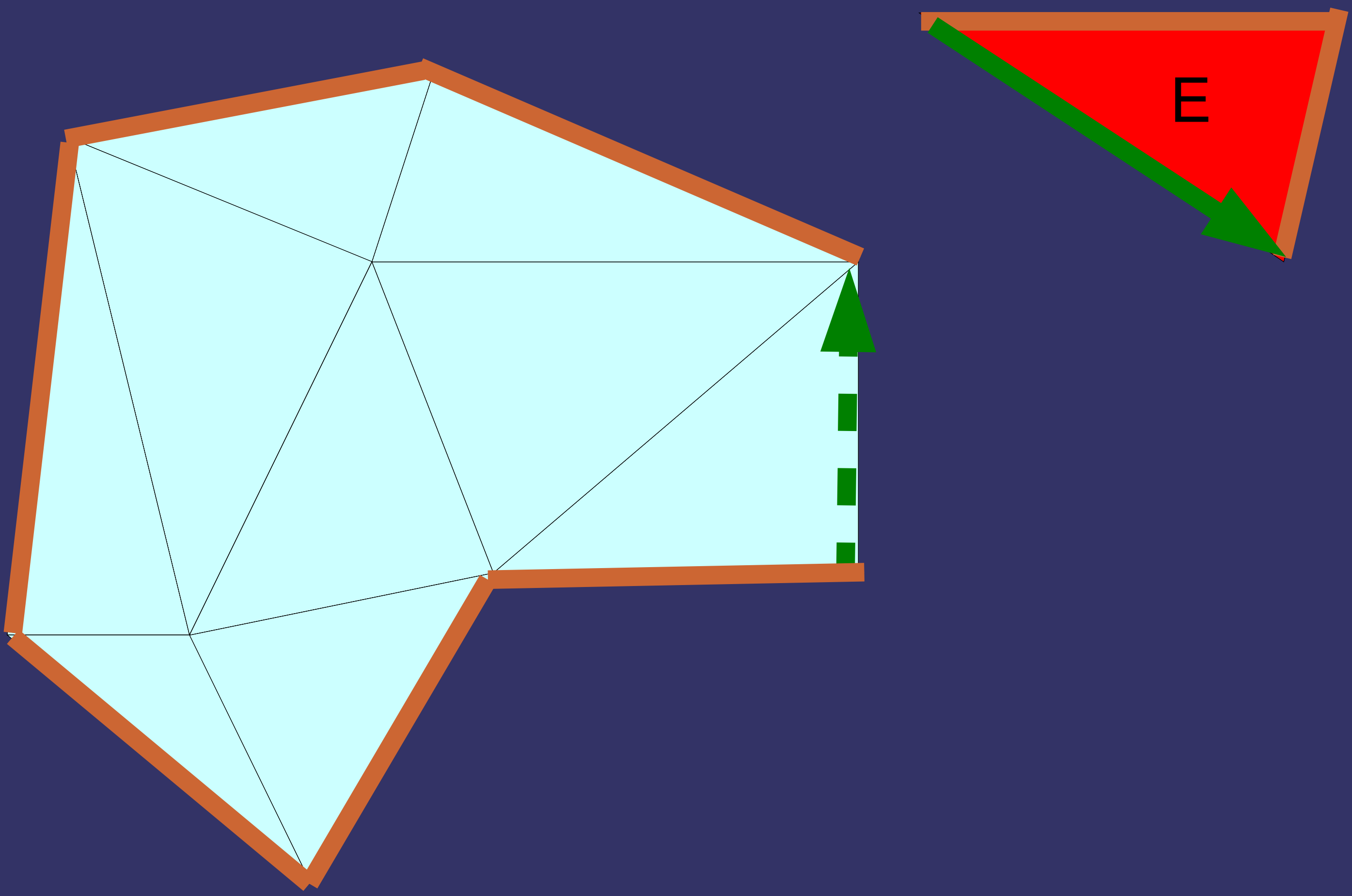
Edgebreaker in action



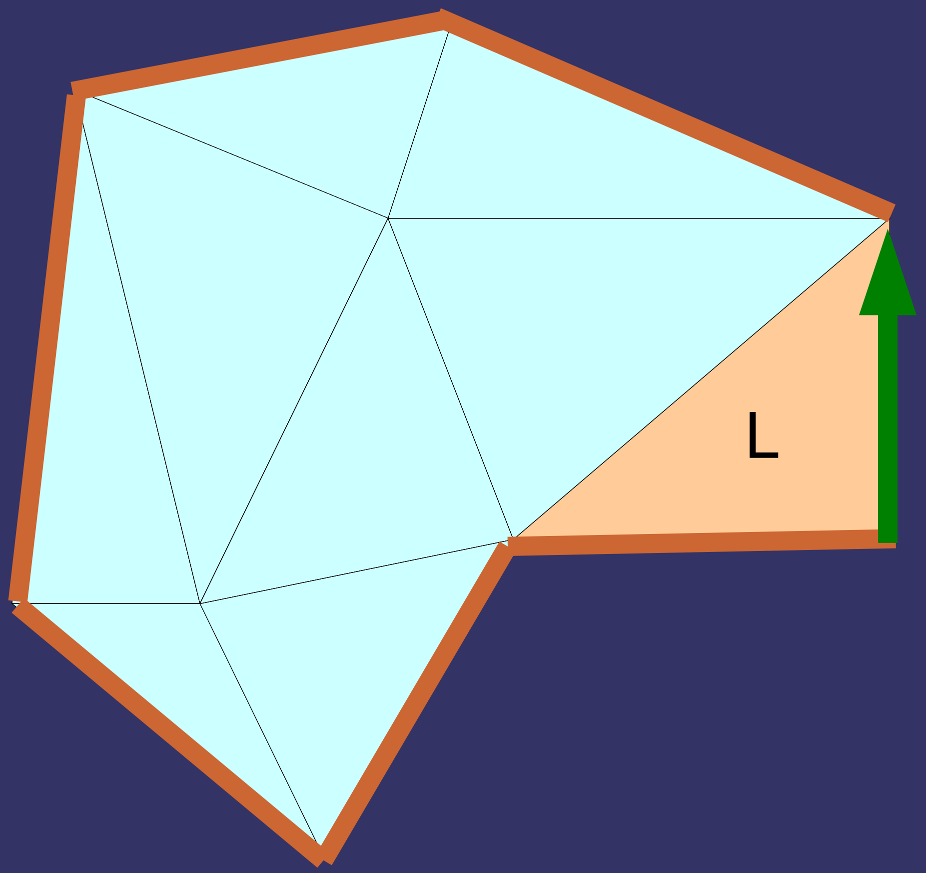
Edgebreaker in action



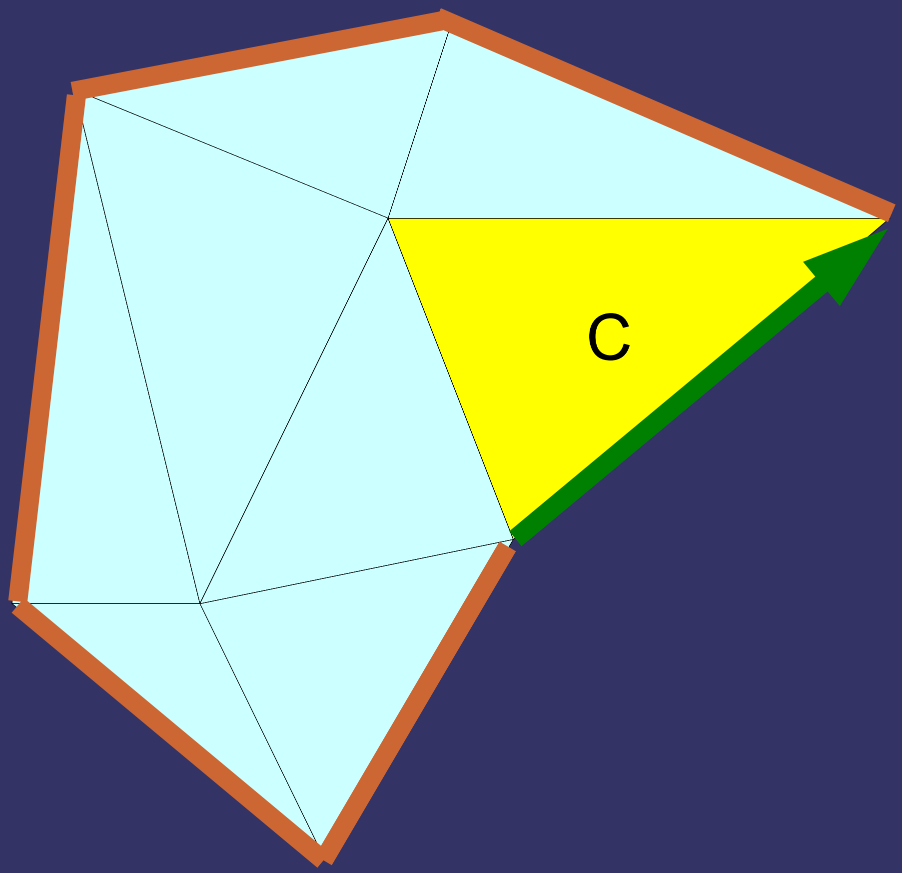
Edgebreaker in action



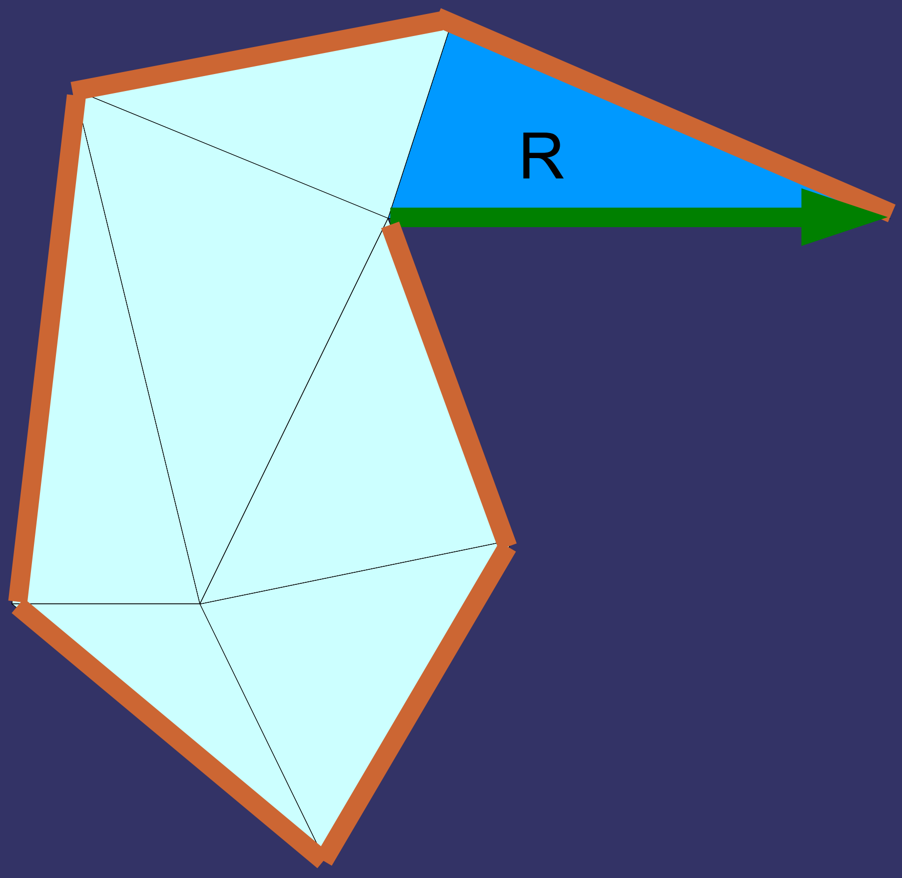
Edgebreaker in action



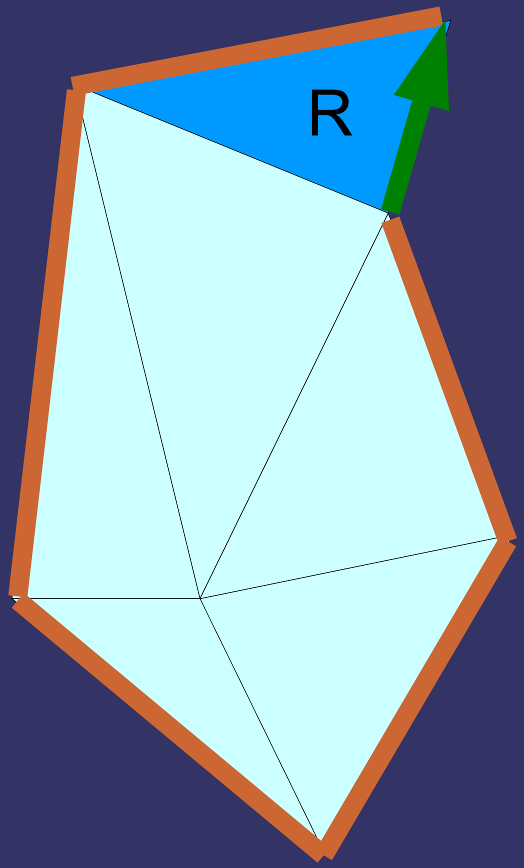
Edgebreaker in action



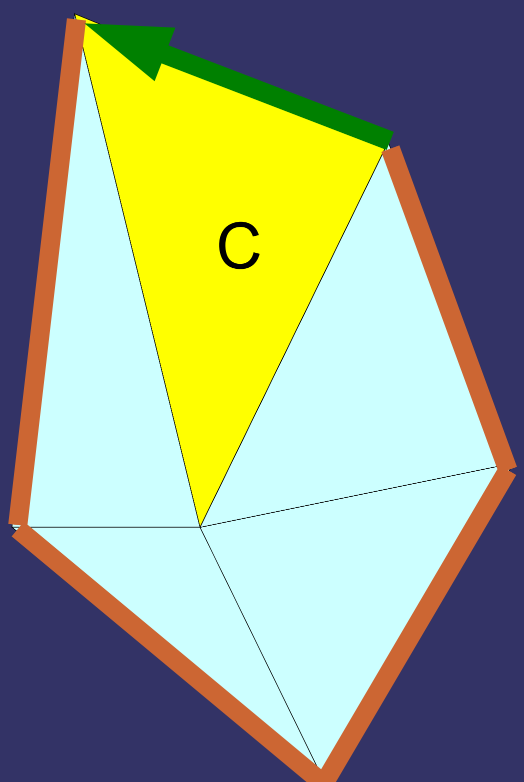
Edgebreaker in action



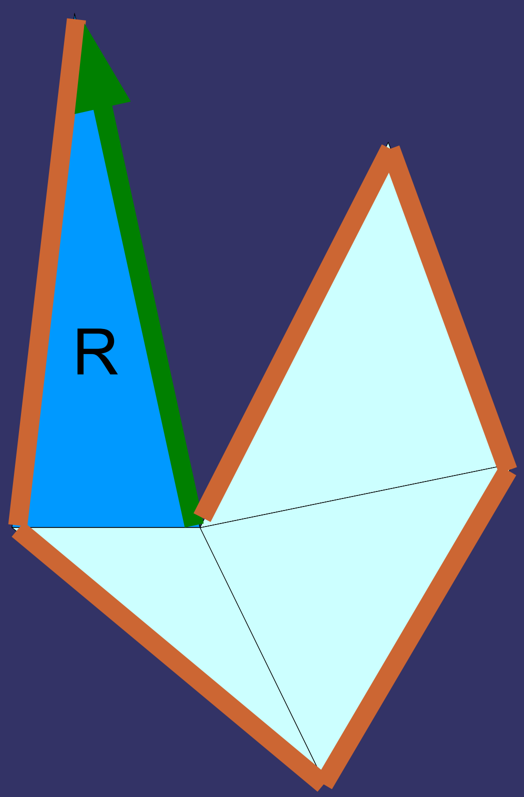
Edgebreaker in action



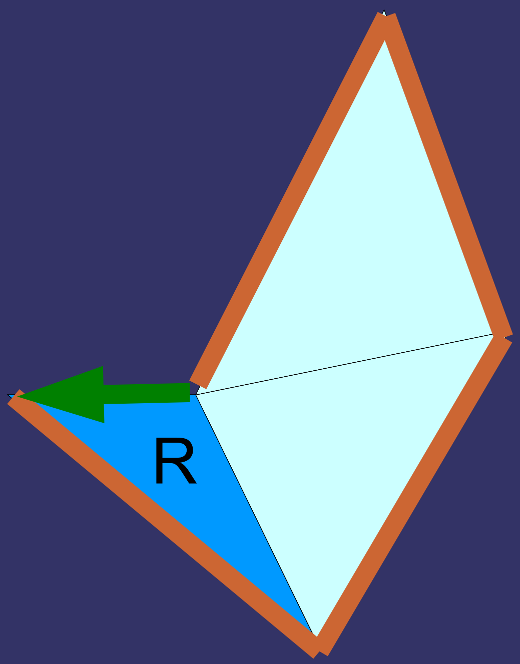
Edgebreaker in action



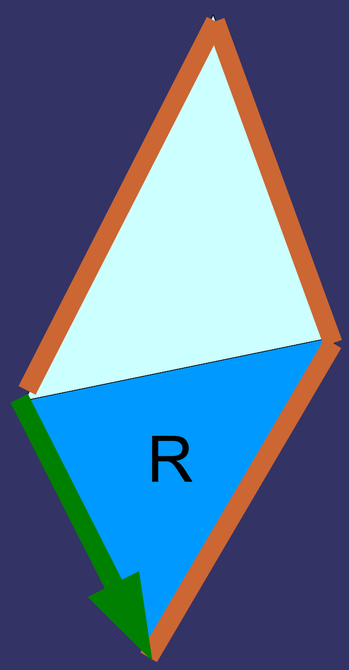
Edgebreaker in action



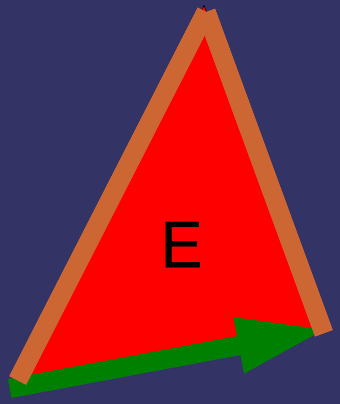
Edgebreaker in action



Edgebreaker in action

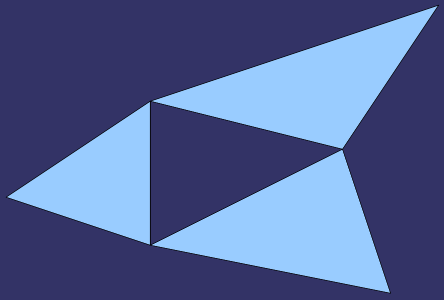


Edgebreaker in action



Terminology

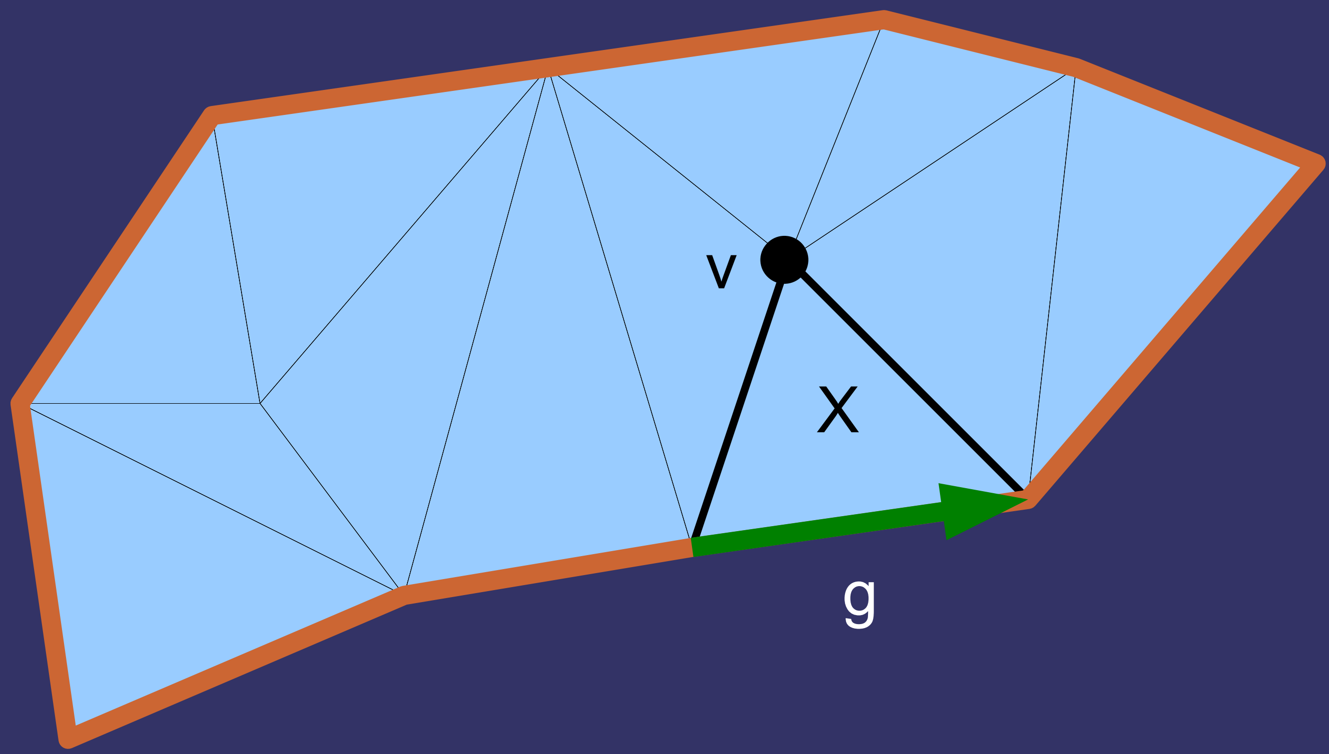
- Simple triangle mesh
  - Edge-connected collection of triangles with one or zero bounding loops
    - Piecewise linear surface homeomorphic to a disk or a sphere
- Hole: Interior triangles missing
  - Like holes in a sheet of paper



- Handle: The surface has genus > 0
  - Doughnuts, teacups etc.

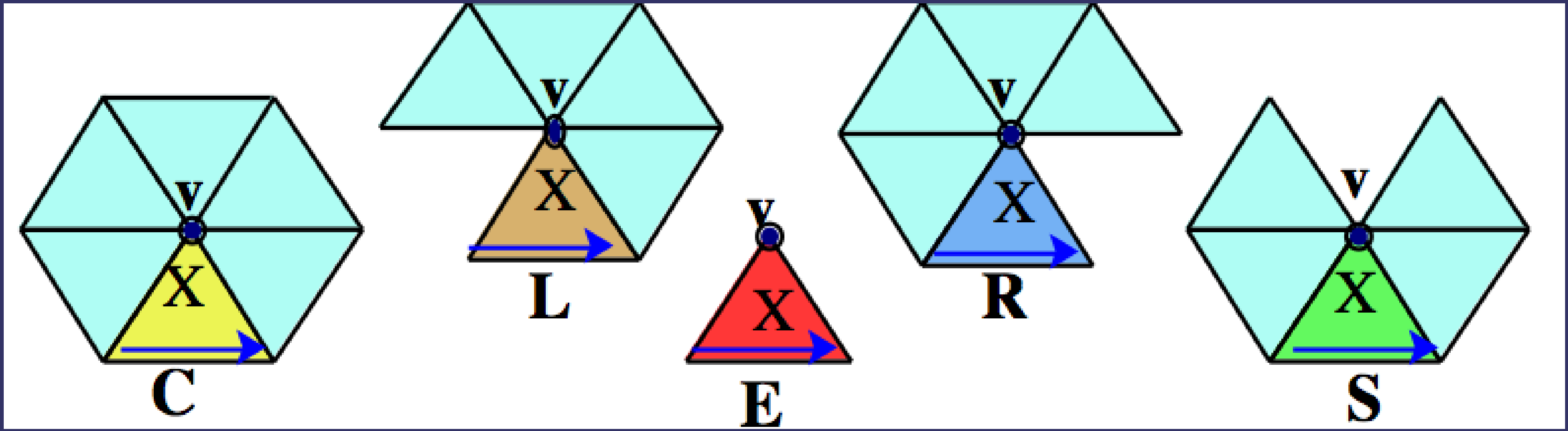
Basics

- Simple triangle mesh T
- “Gate” g := some half-edge on bounding loop
- X := triangle incident on g, v := third vertex of X



Types of Triangles

- The g-X-v combo can be one of the 5 types C L E R S

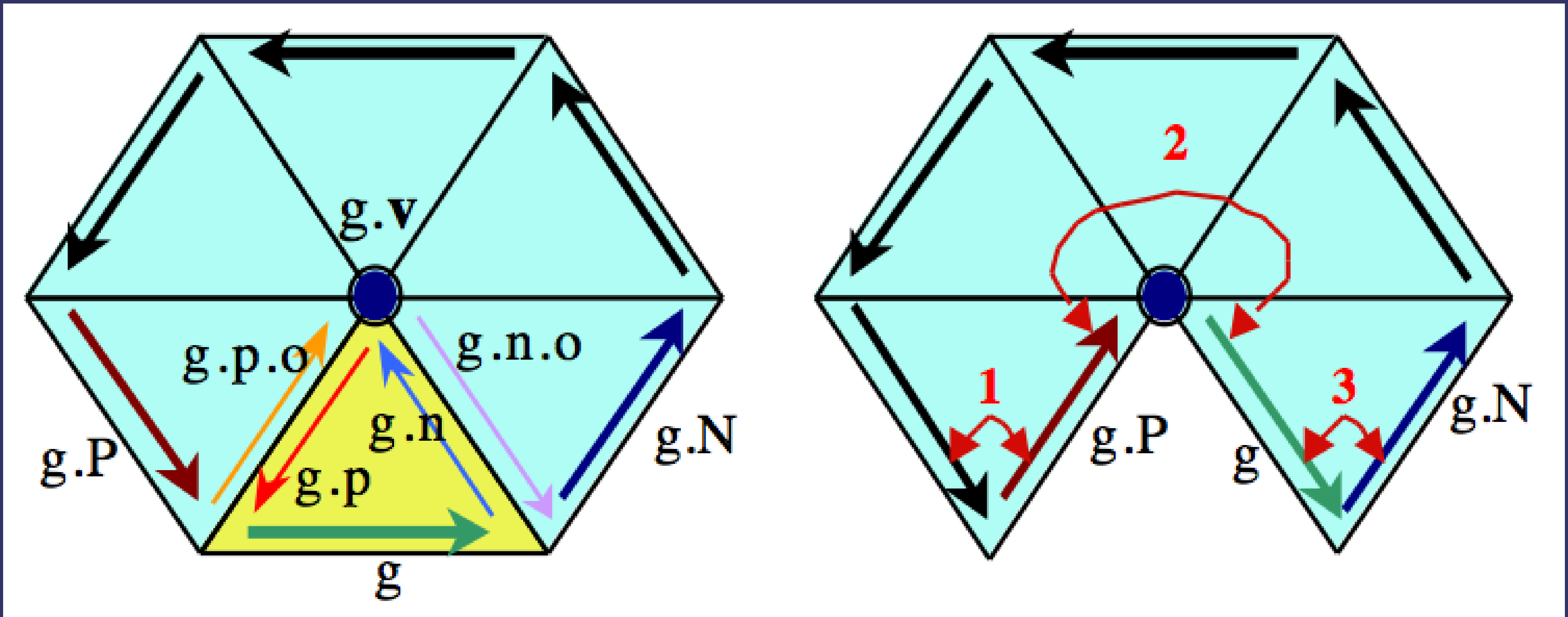


- Compression scheme: remove X, store the operation (C, L, E, R or S), update the bounding loop and advance the gate

Data Structures

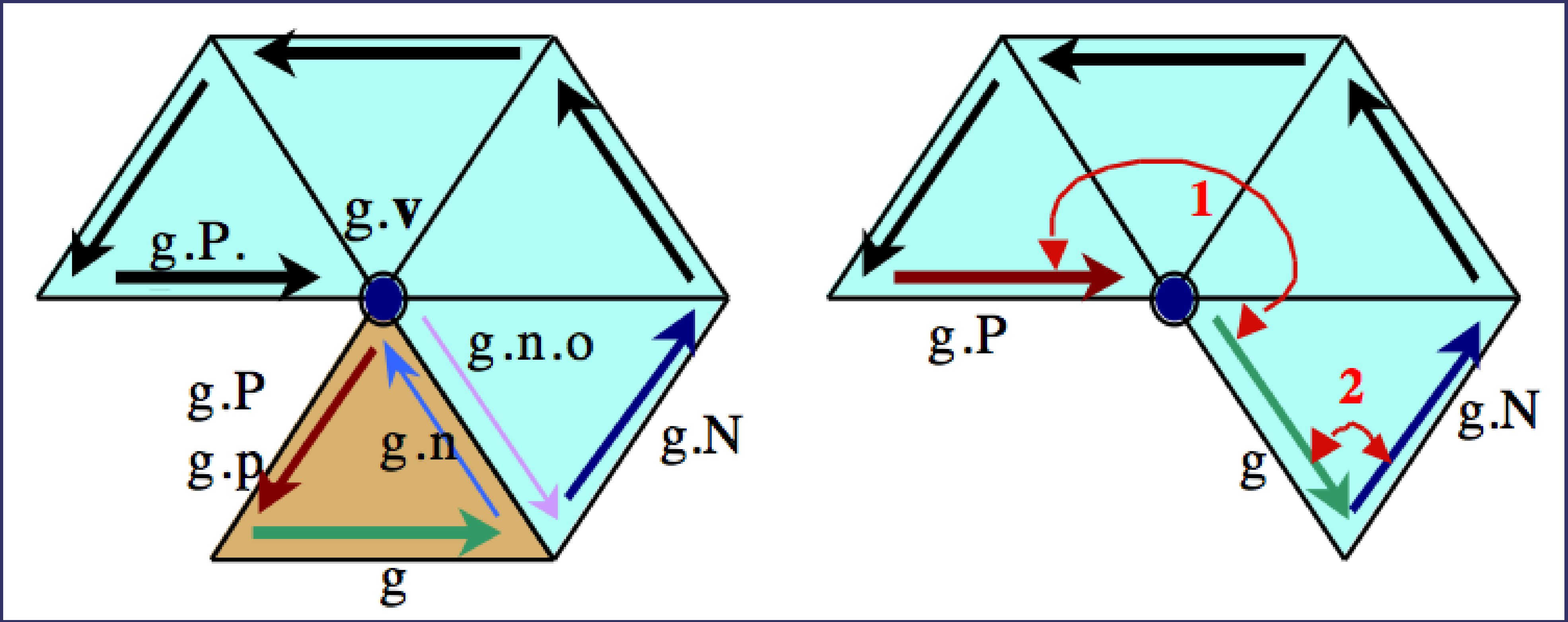
- Mesh: (pseudo) winged-edge structure
- History H: stores CLERS opcodes
- List P: stores vertex positions
- Stack: stores “deferred” loops

Operation C (central)

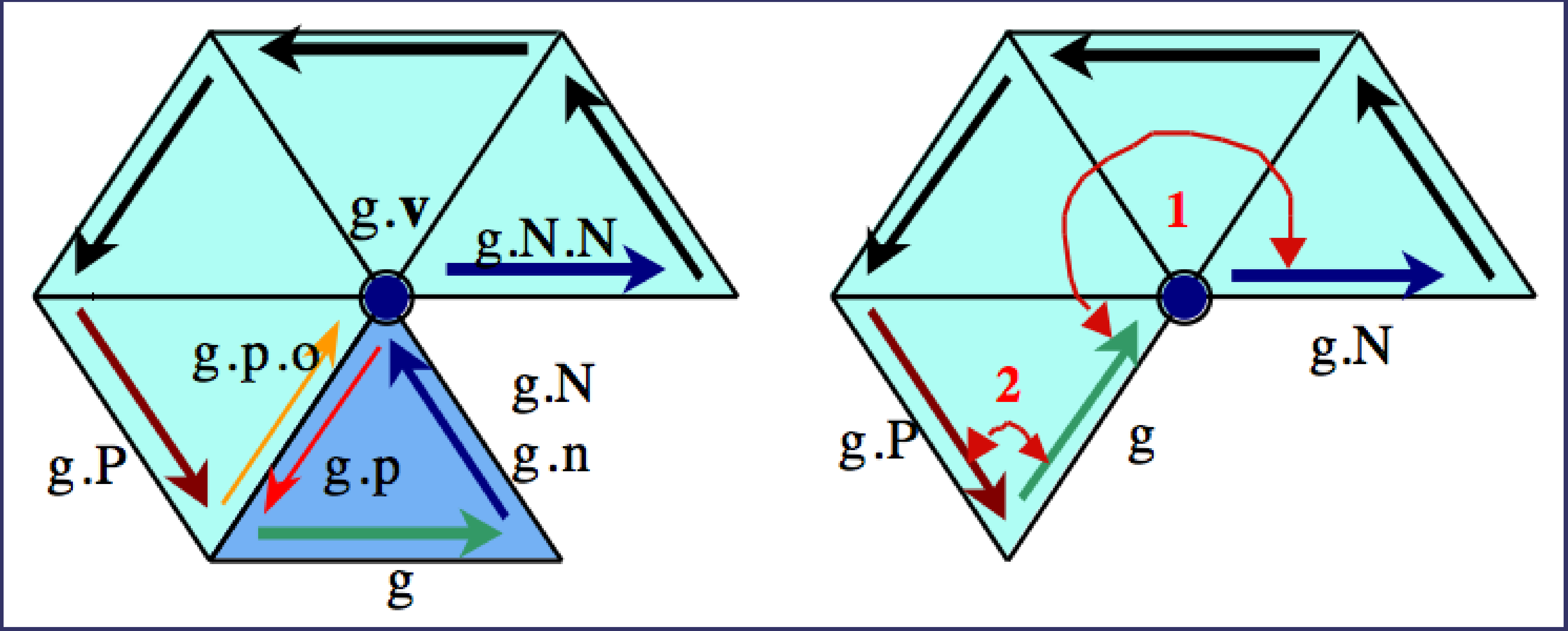


- v == internal vertex before the op
  == boundary vertex after the op
- Position of v is appended to list P

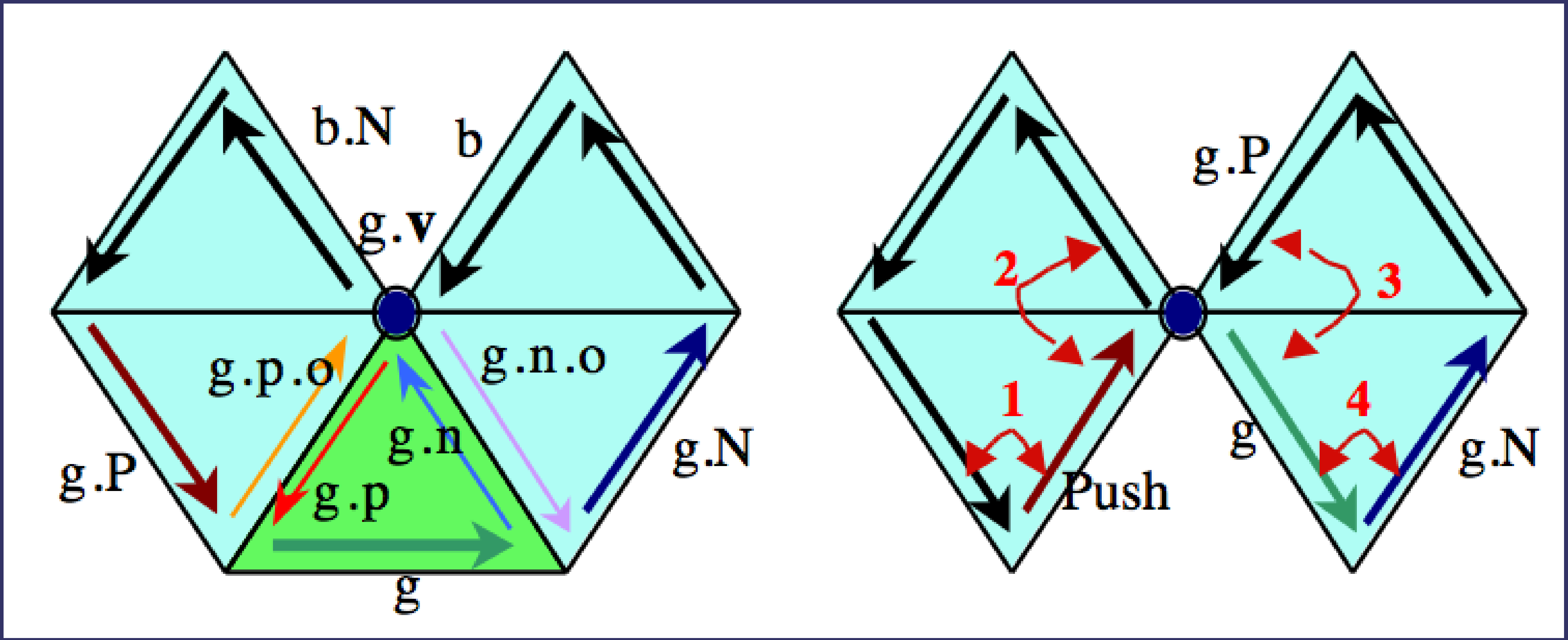
Operation L (left)



Operation R (right)

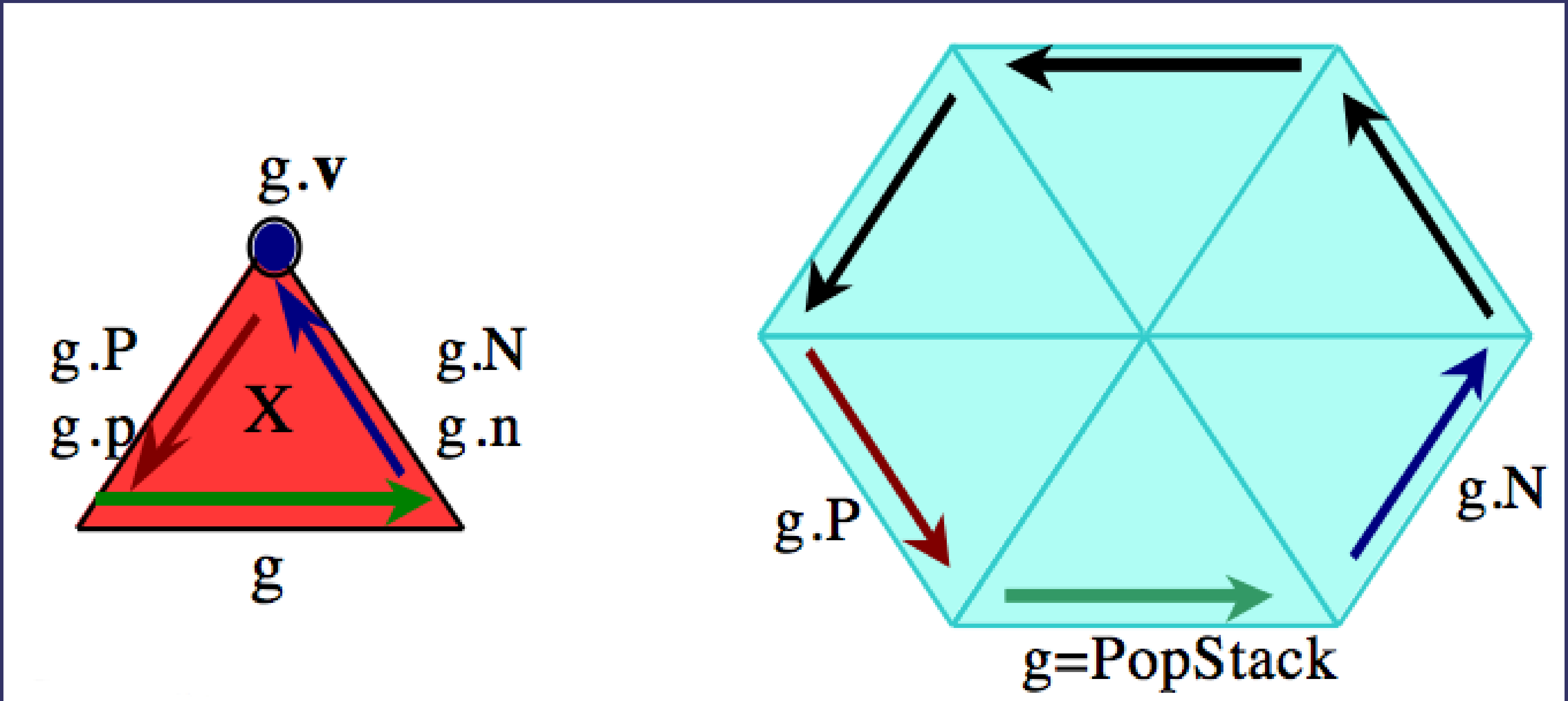


Operation S (split)



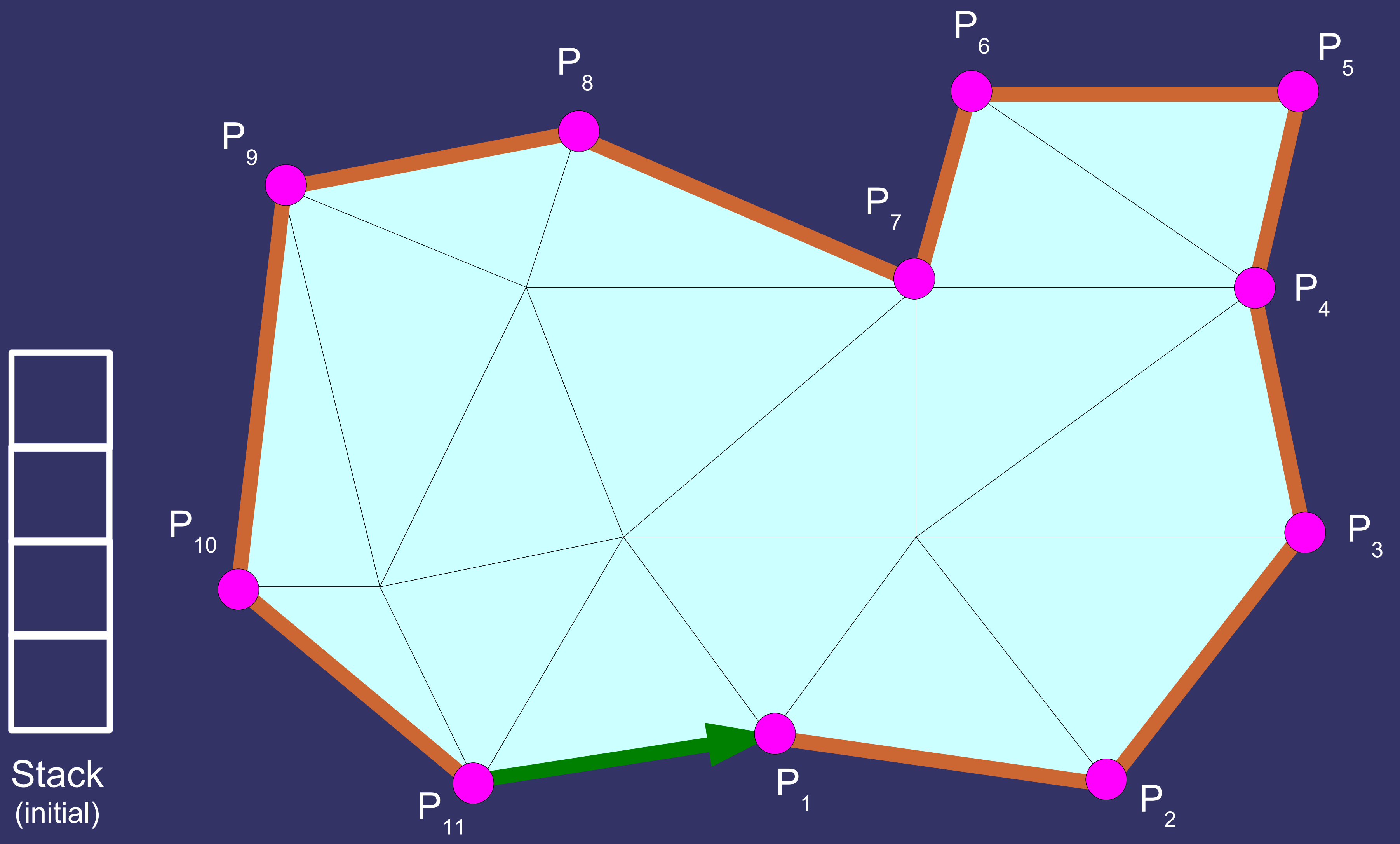
- Two loops created.
- Push gate of one loop onto the stack and continue with the other loop

Operation E (end)



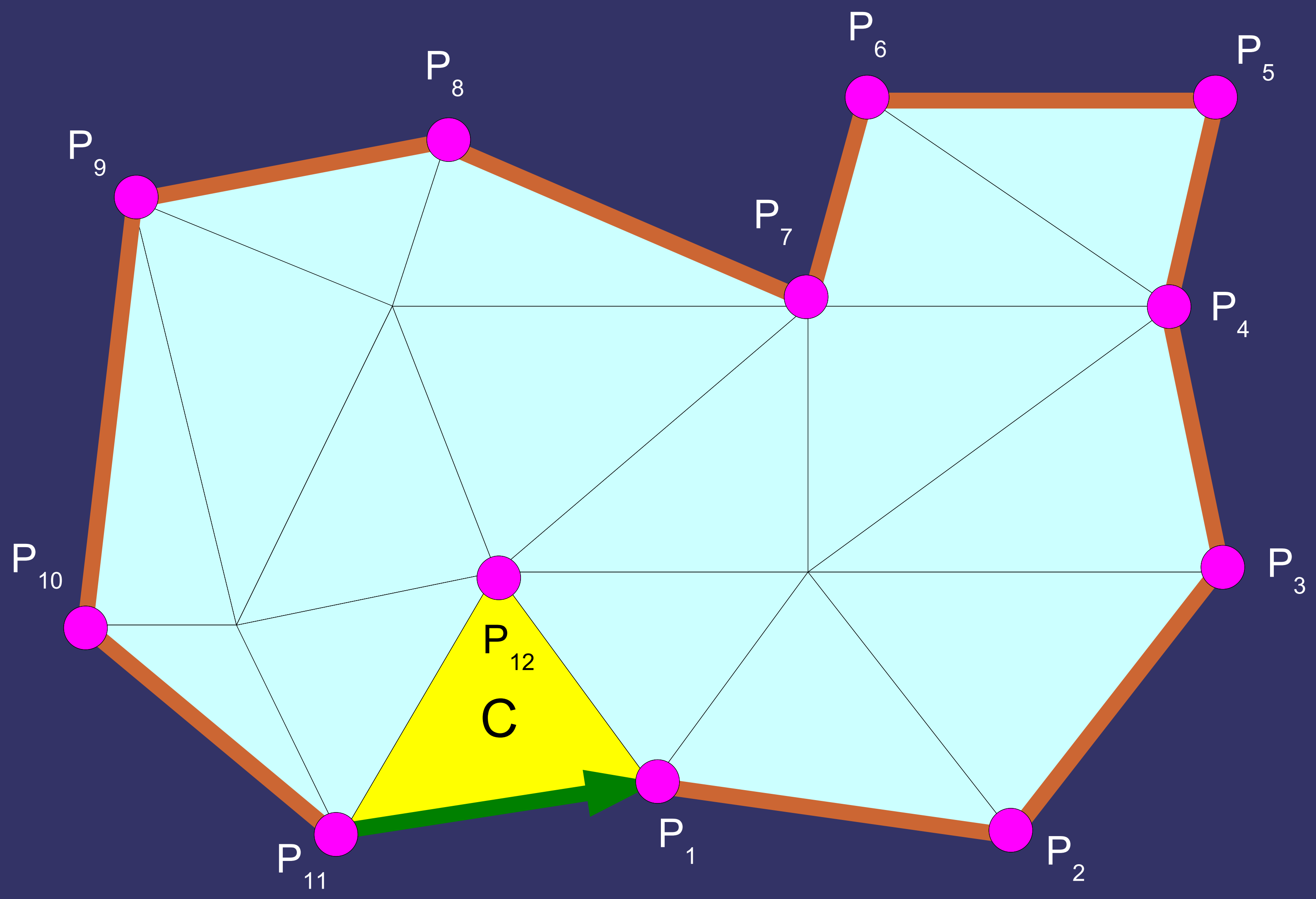
- Loop shrinks to zero
- Pop the gate of the next loop to be processed from the stack

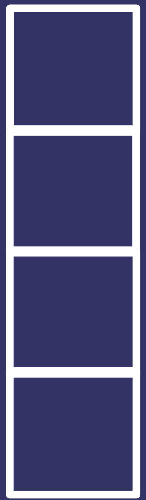
Edgebreaker in action



History = (null)

Edgebreaker in action

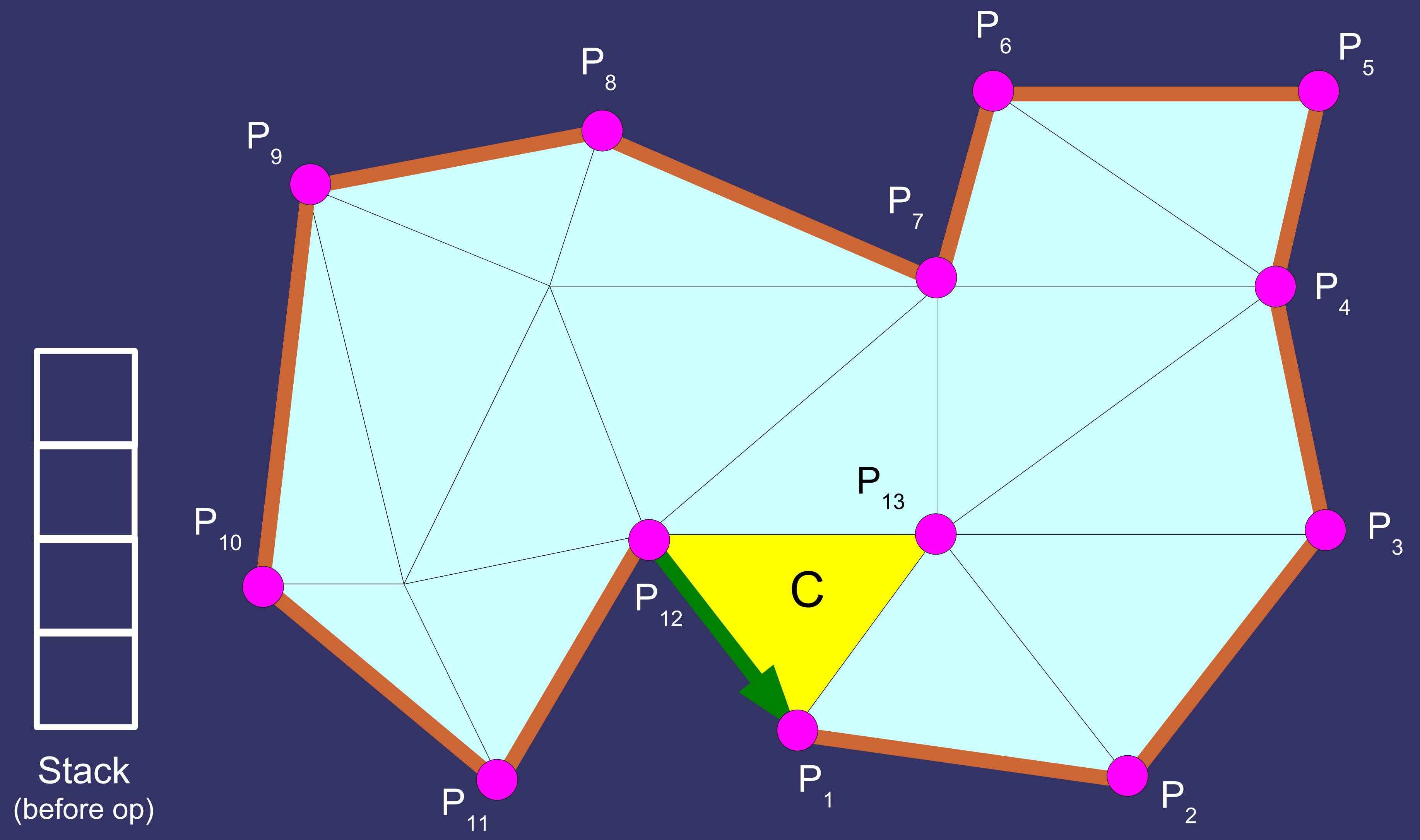




Stack
(before op)

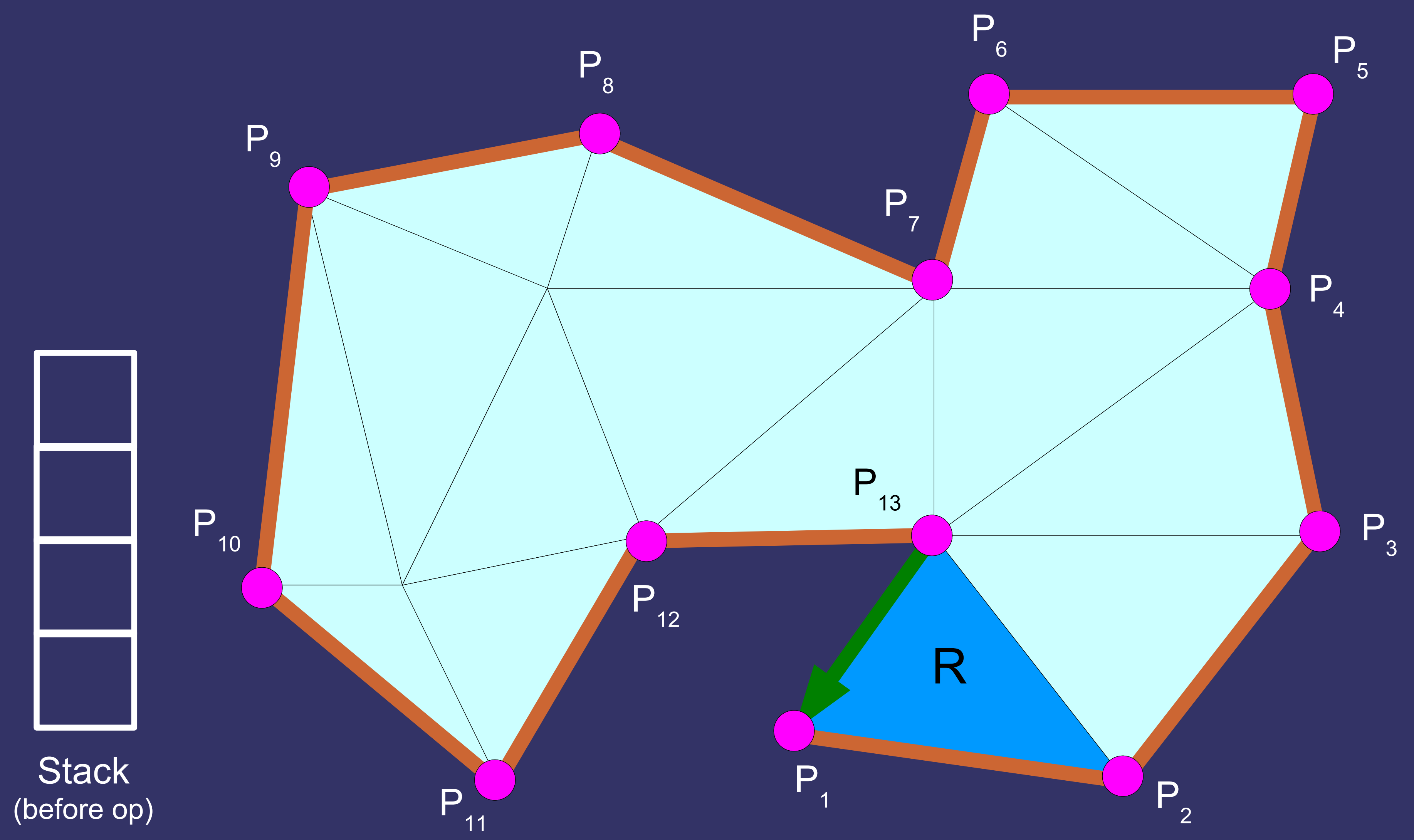
History = C

Edgebreaker in action



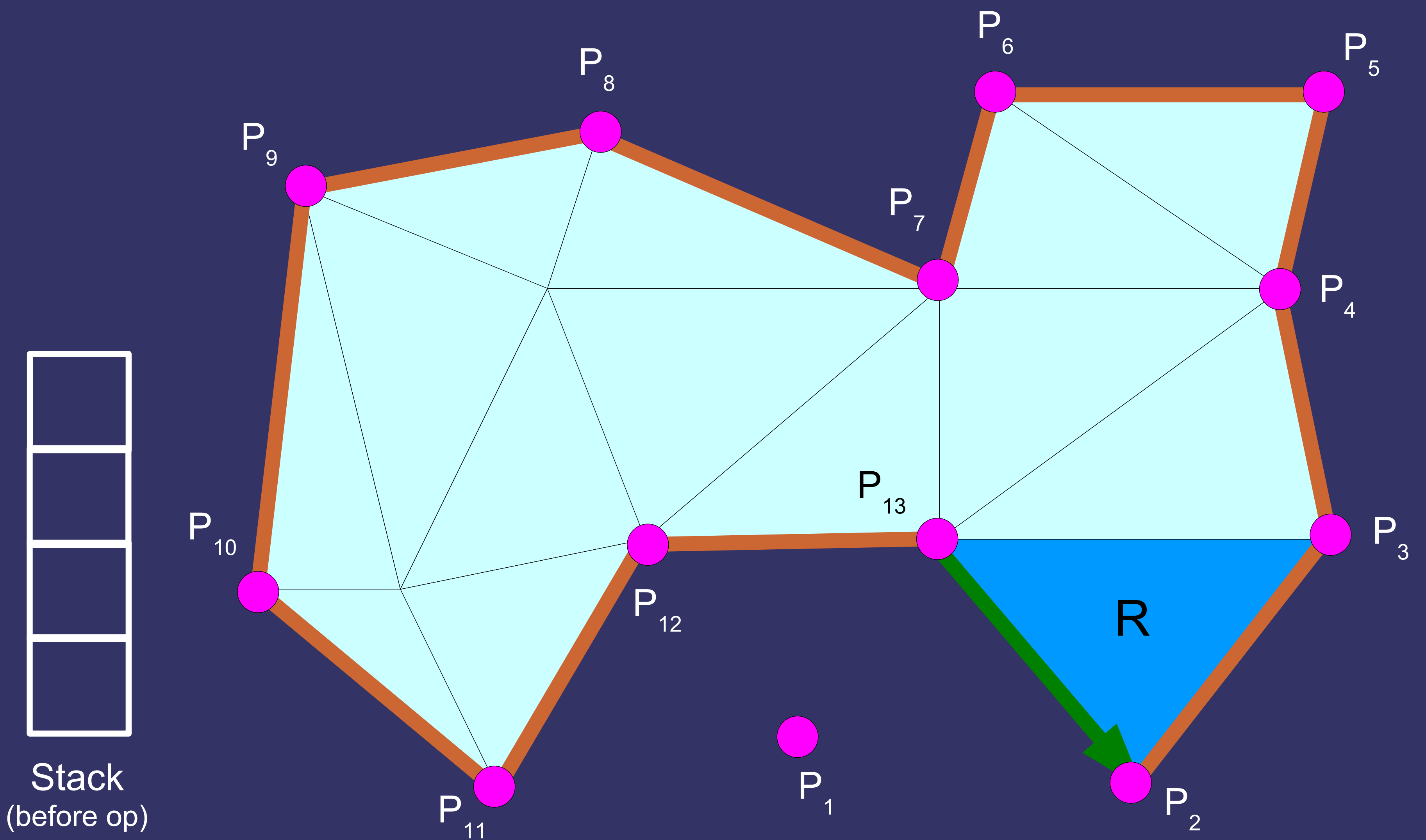
History = CC

Edgebreaker in action



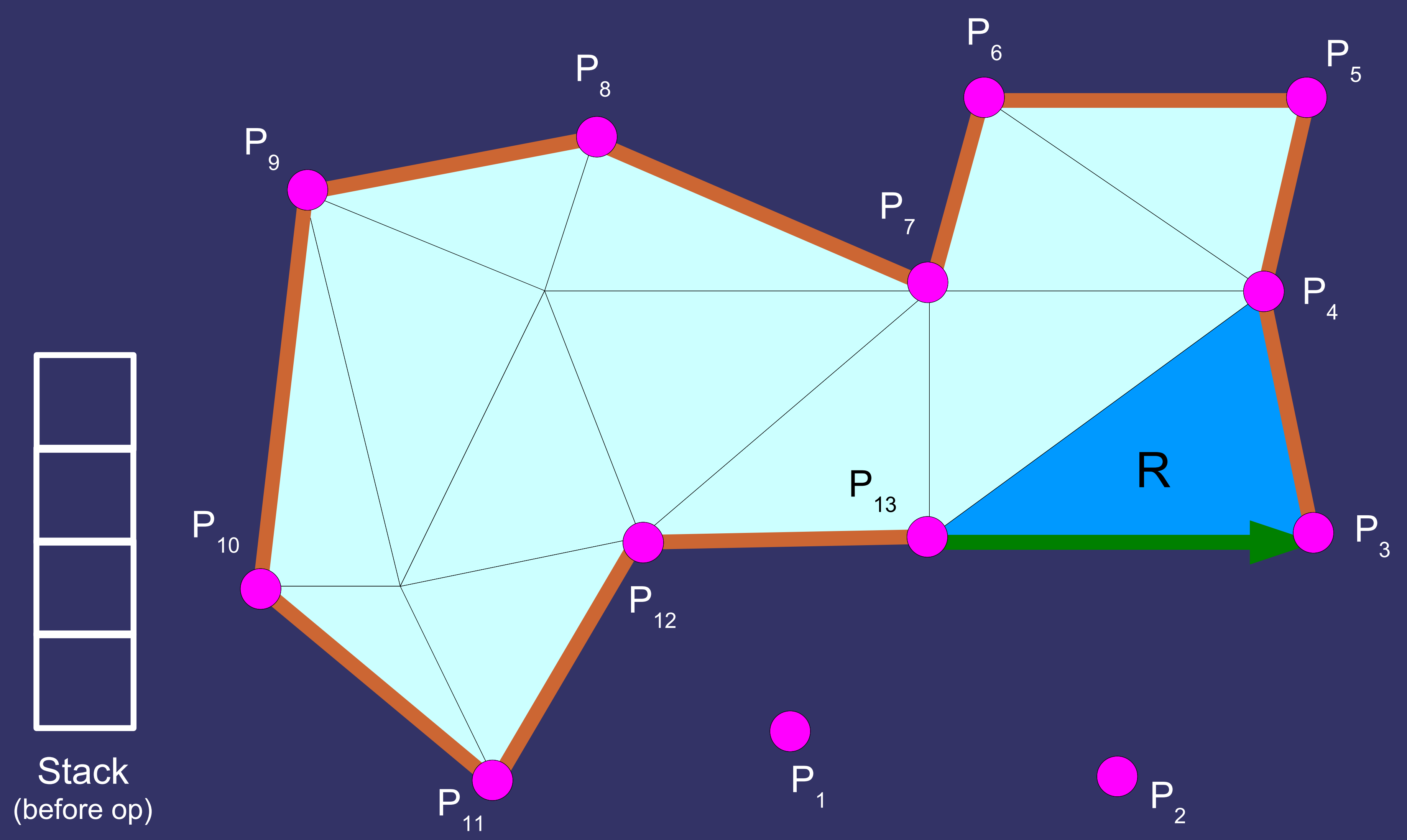
History = CCR

Edgebreaker in action



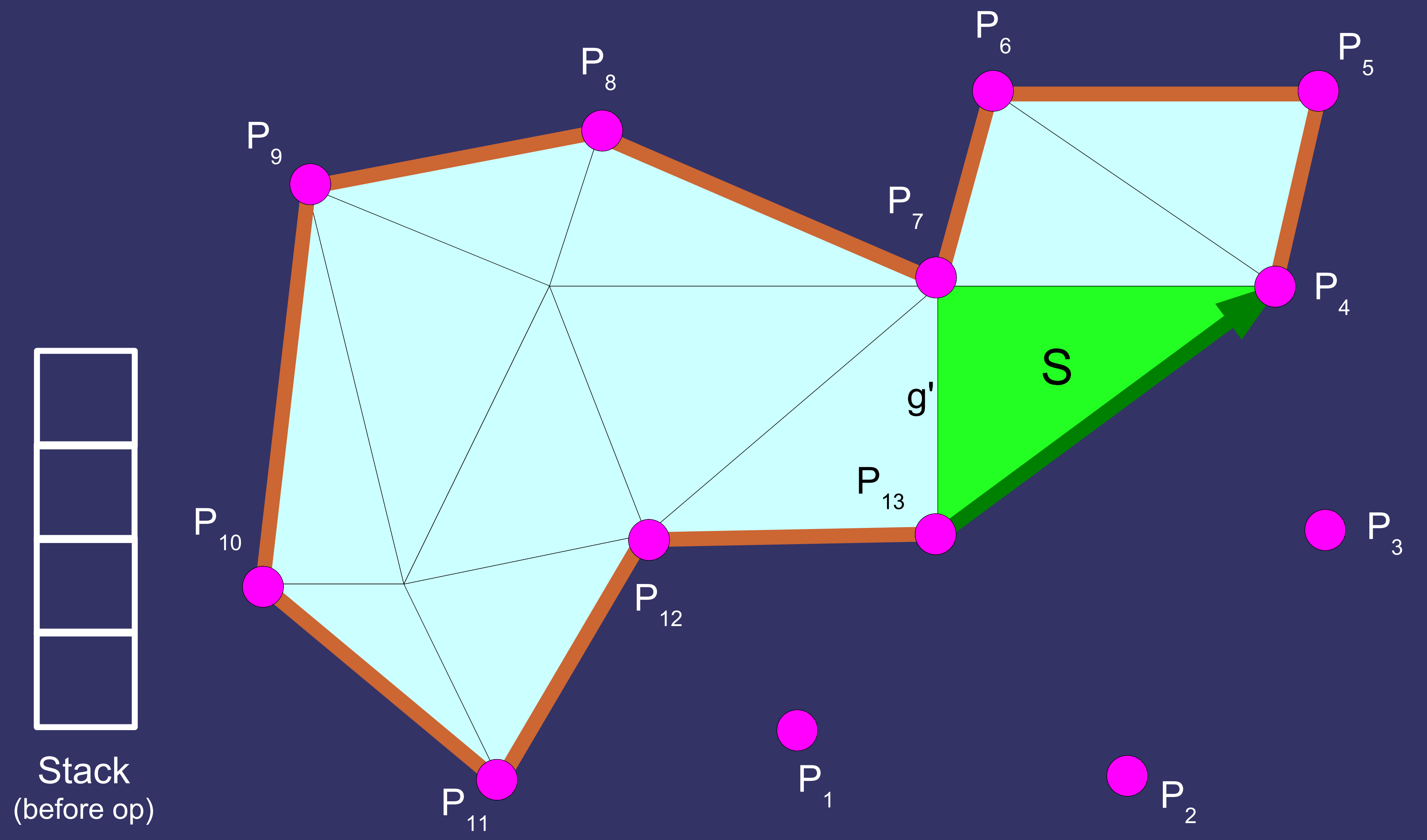
History = CCRR

Edgebreaker in action



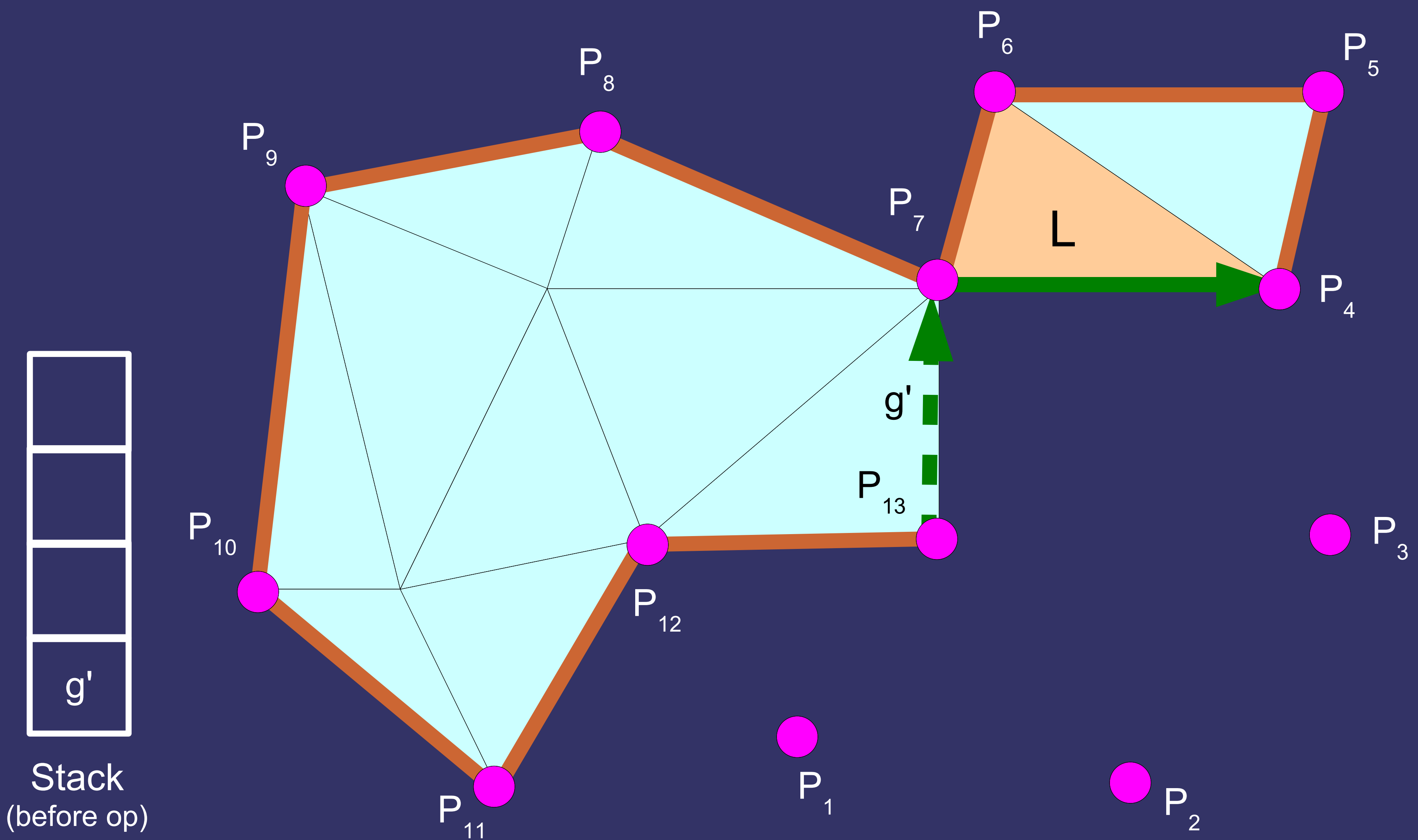
History = CCRRR

Edgebreaker in action



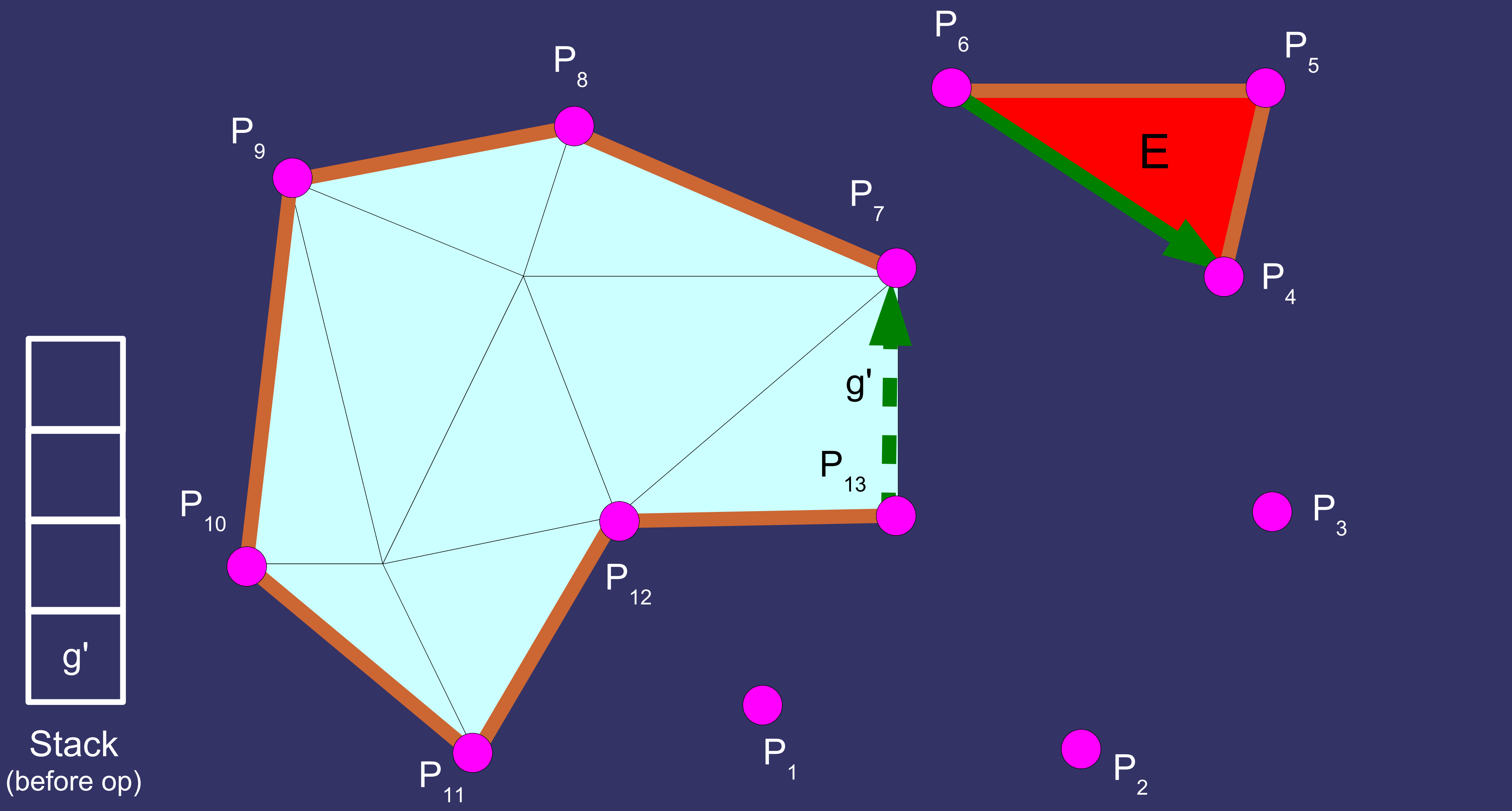
History = CCRRRS

Edgebreaker in action



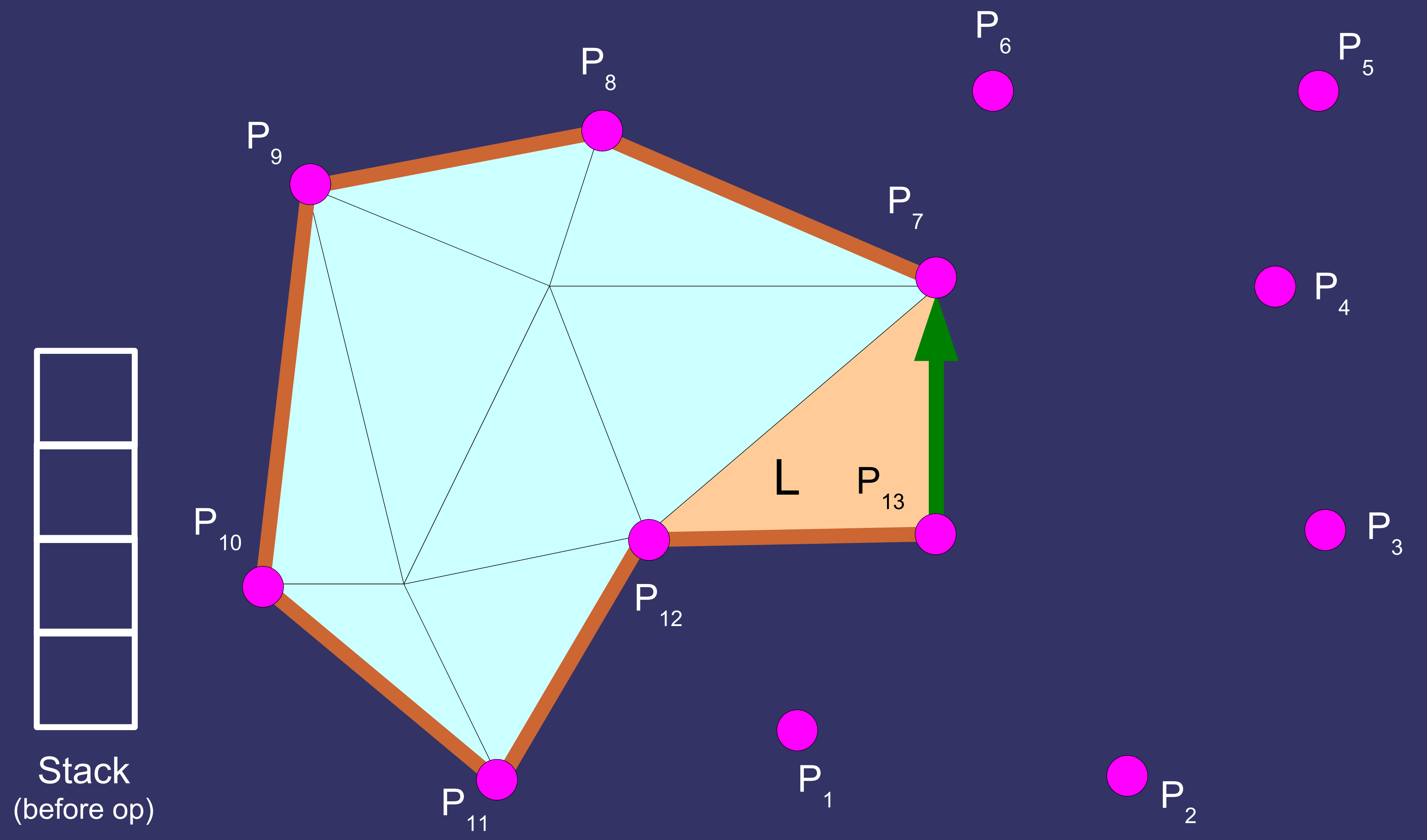
History = CCRRRSL

Edgebreaker in action



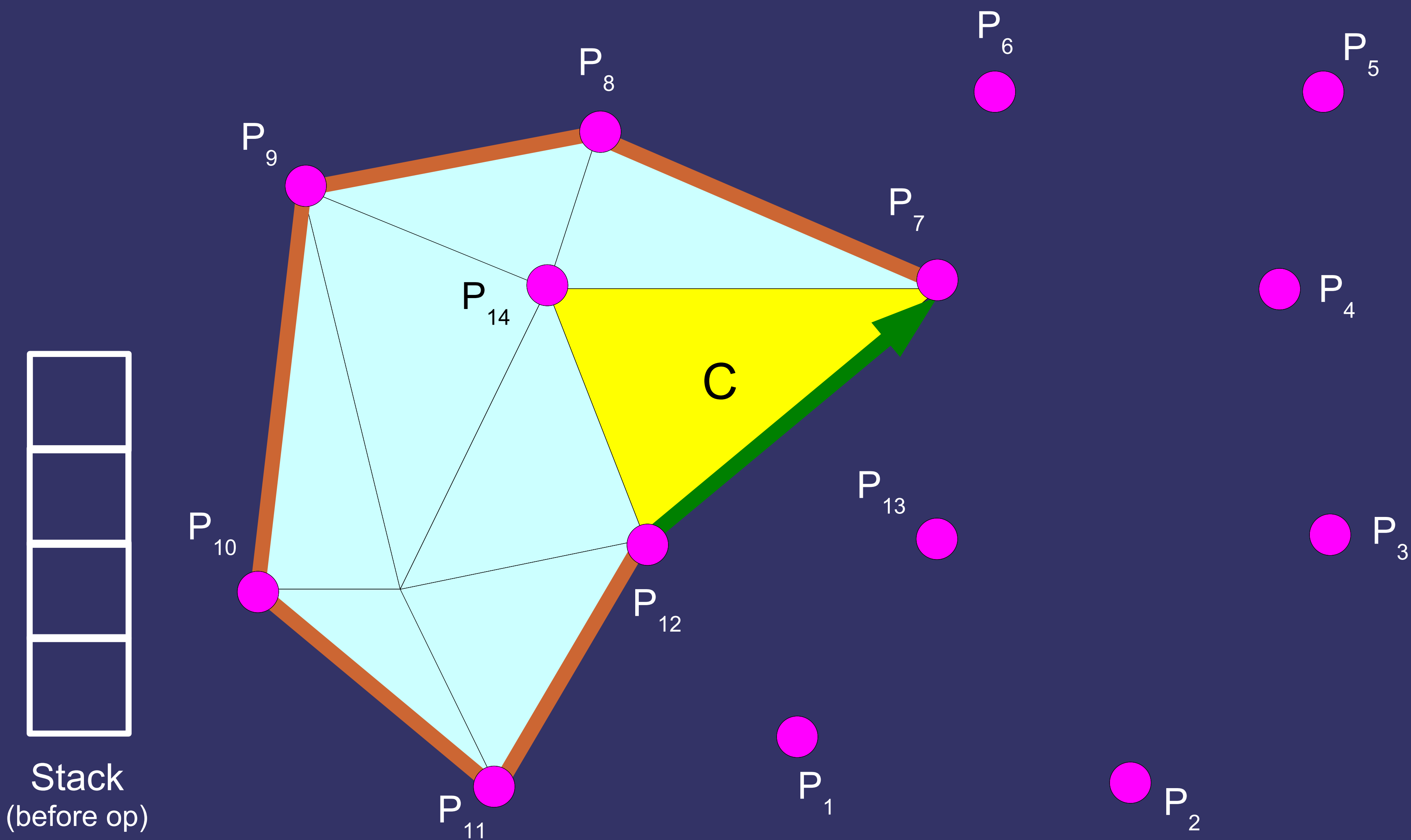
History = CCRRRSLE

Edgebreaker in action



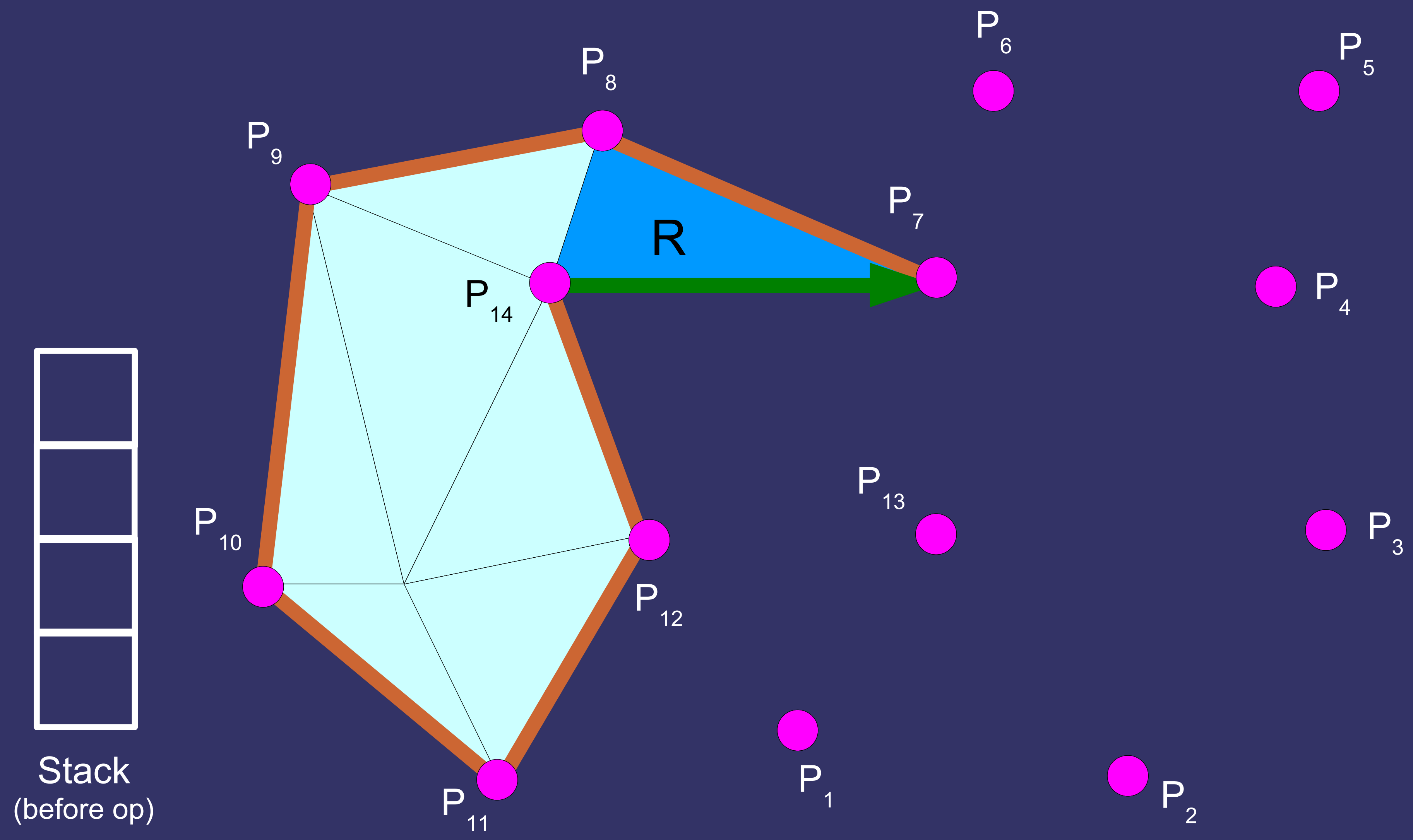
History = CCRRRSLEL

Edgebreaker in action



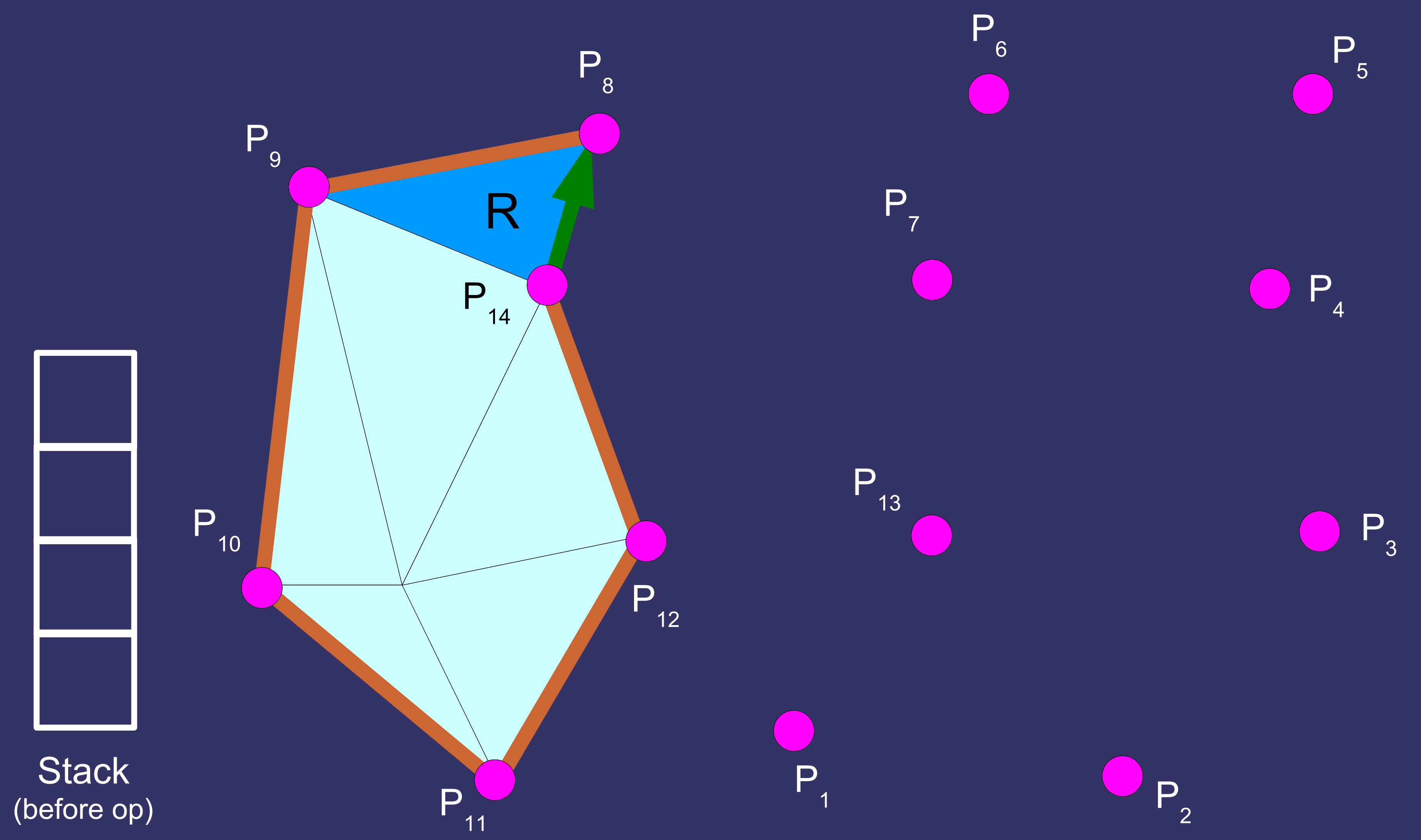
History = CCRRRSLELC

Edgebreaker in action



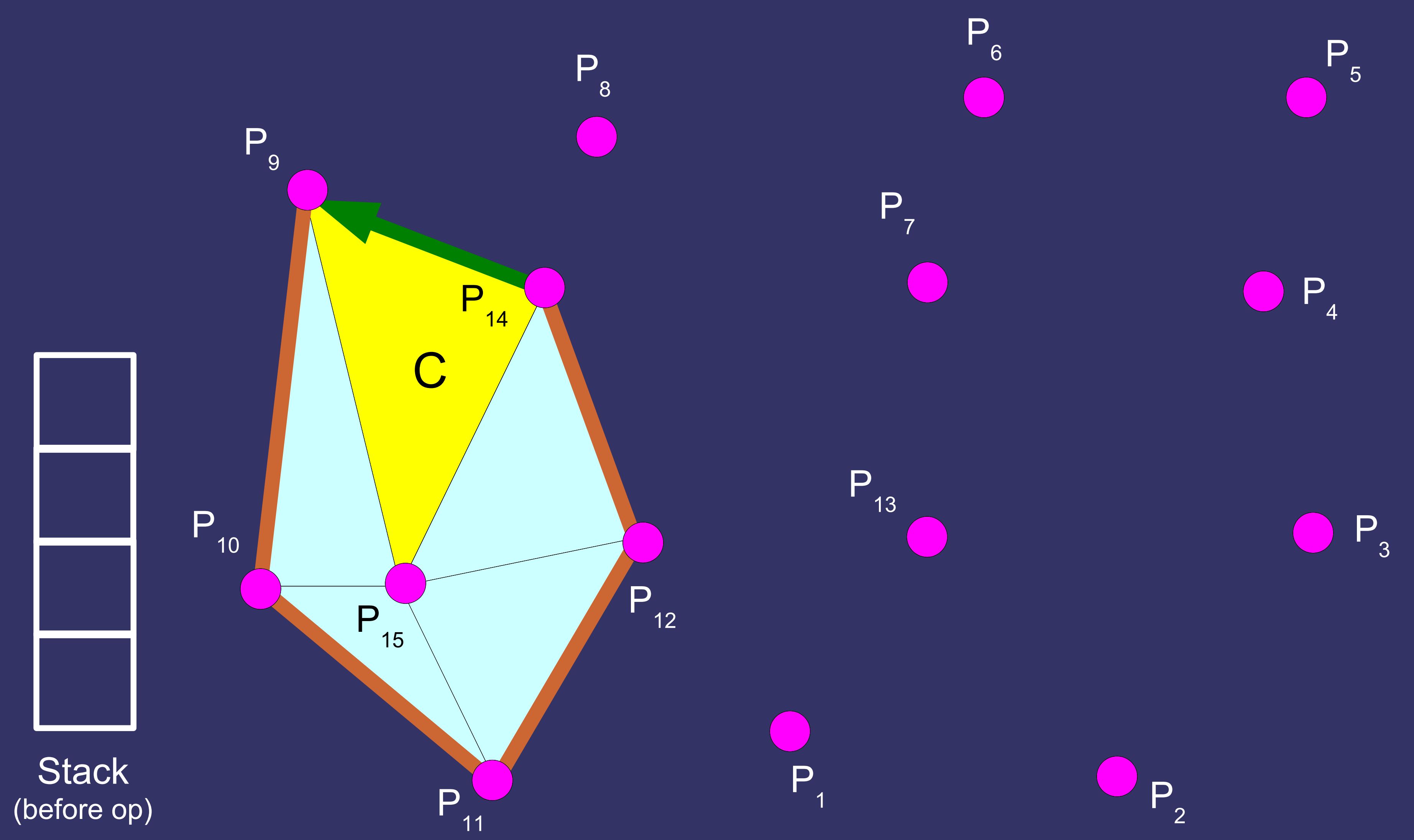
History = CCRRRSLELCR

Edgebreaker in action



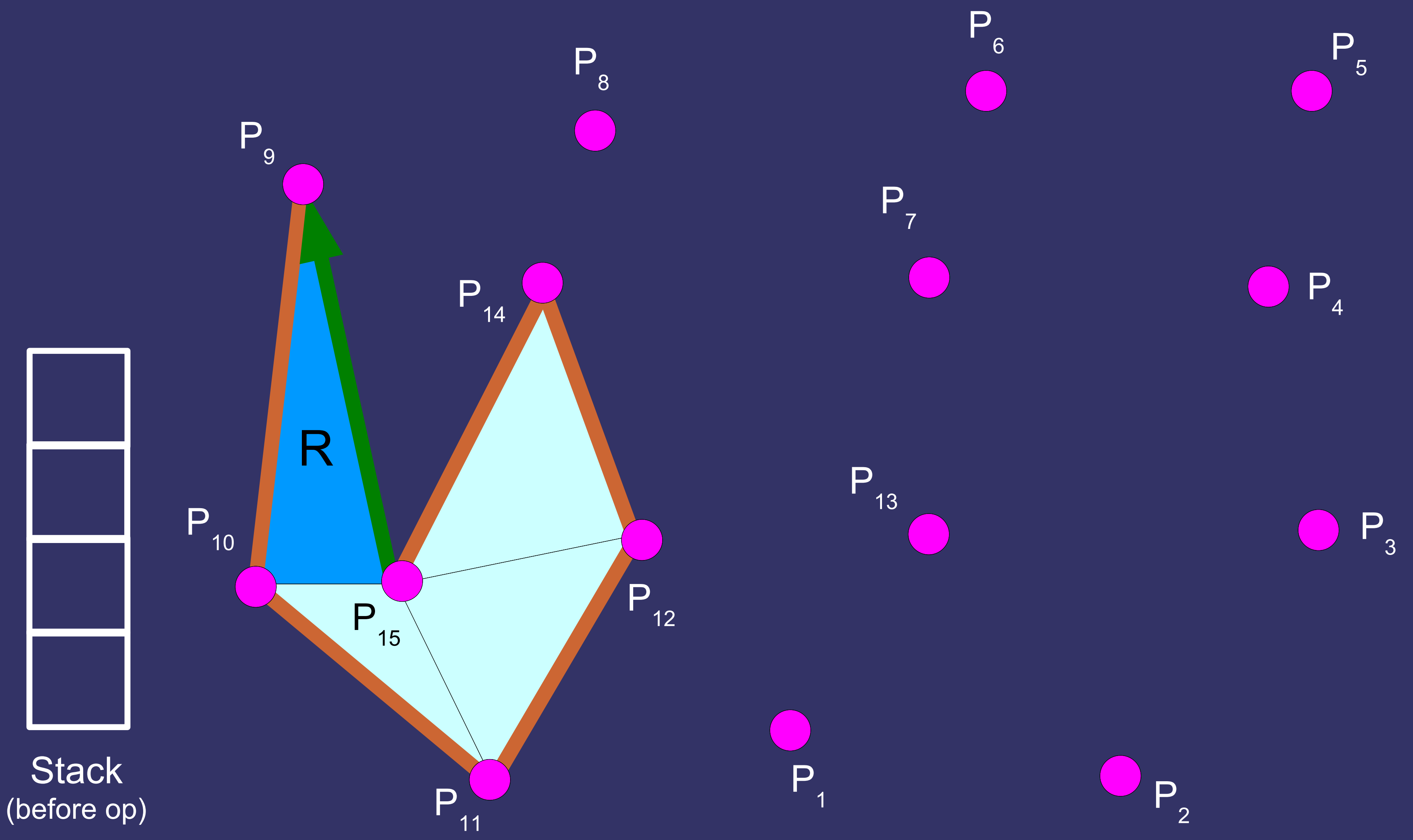
History = CCRRRSLELCRR

Edgebreaker in action



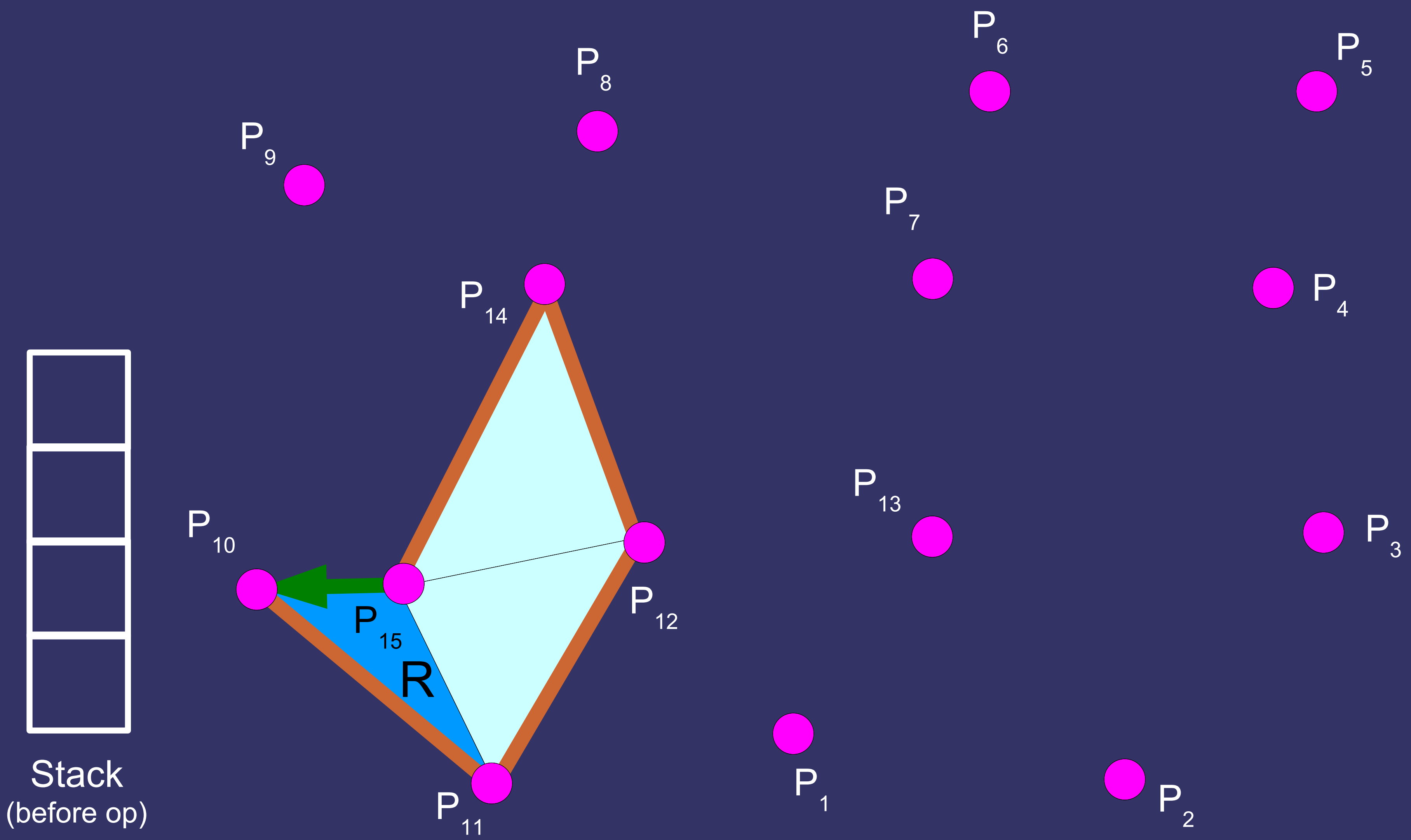
History = CCRRRSLELCRRC

Edgebreaker in action



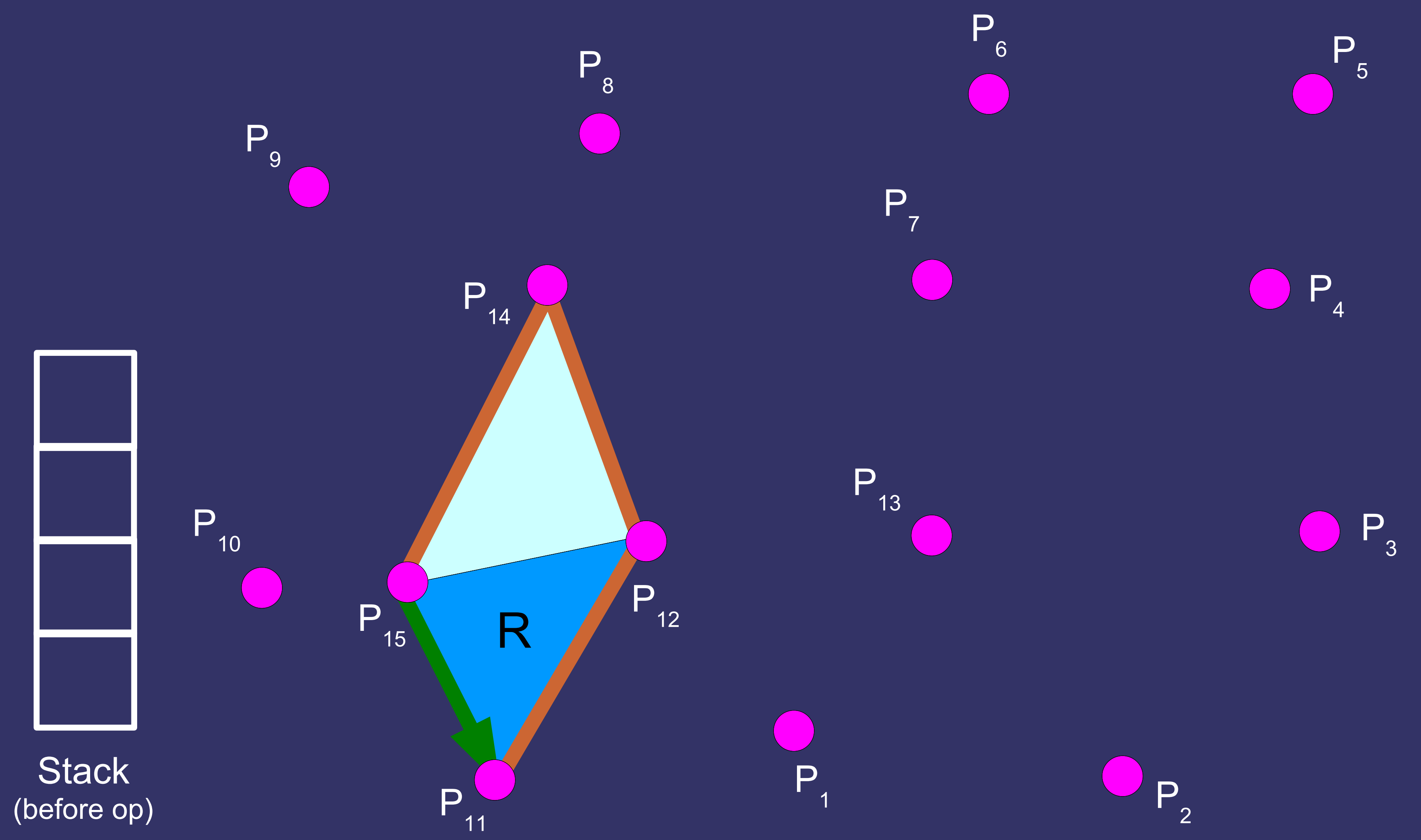
History = CCRRRSLELCRRCR

Edgebreaker in action



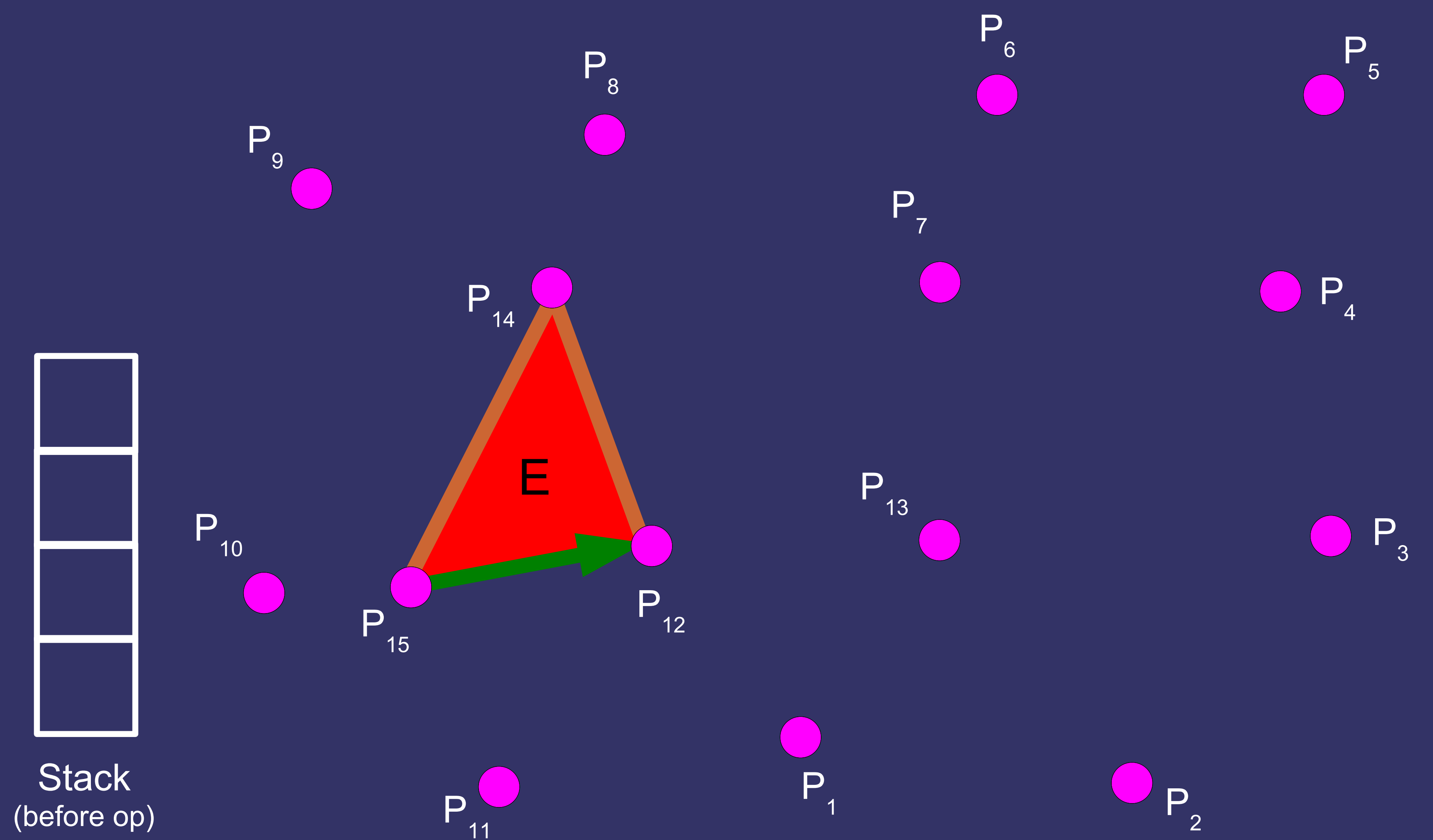
History = CCRRRSLELCRRCRR

Edgebreaker in action



History = CCRRRSLELCRRCRRR

Edgebreaker in action



History = CCRRRSLELCRRCRRRE

Crux Observations

- Different graphs have different histories.
  - Allows reconstruction of topology.

- There is a bijection between C operations and interior vertices of the triangulation.
  - Allows reconstruction of embedding.

Compression

- The history H can be encoded as follows:
  - Write 0 for C
  - Write 100 for S
  - Write 101 for R
  - Write 110 for L
  - Write 111 for E
- (The resulting binary string may be further compressed using any good compression algorithm.)

Compression

- Number of bits required = b
  $= |C| + 3(|S| + |L| + |R| + |E|)$
- Denote boundary vertices V_E and interior vertices V_I.
- $|T| = |C| + |S| + |L| + |R| + |E|$ (each op destroys one triangle)
- $|C| = |V_I|$ (observe from algorithm)
- $|T| = 2|V_I| + |V_E| - 2$ (property of triangulations)
- Hence $b = 2|T| + |V_E| - 2 \leq 3|T|$

Compression

- Assume mesh has small boundary.
  - e.g., a *closed* genus 0 surface can be "opened up" by "cutting along" one of its edges; the resulting surface has a boundary of length 2.
- Then $b \approx 2|T|$, i.e. 2 bits per triangle.
  - Compact repr. of any planar triangulated graph!
- Since CL and CE sequences are impossible, encoding can be made even shorter.
- In other situations, different coding schemes may be used, all of which guarantee $b \approx 2|T|$.

Compression

- Improvement in guarantees:

  [King-Rossignac '99] show coding schemes which guarantee 1.84|T| length for the encoded history of closed genus 0 meshes.

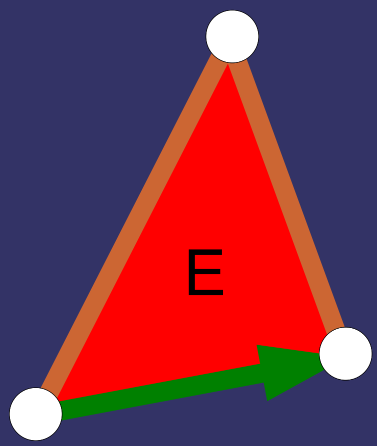
- In practice:

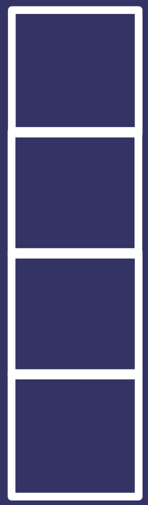
  [Rossignac-Szymczak '99] show that entropy codes “usually” give 0.91|T| to 1.26|T| lengths.

Decompression

- [Rossignac '99] proposes (somewhat convoluted) $O(|T|^2)$ algorithm.
  - Traverses history in compression order, uses preprocessing pass to compute constants + offsets and generation pass to actually recreate the graph.
- [Rossignac-Szymczak '99] propose $O(|T|)$ "Wrap&Zip" algorithm.
- [Isenburg-Snoeyink '01] propose single-pass $O(|T|)$ algorithm.
  - Traverses history in reverse order (why did it take two years to devise this "obvious" scheme???)

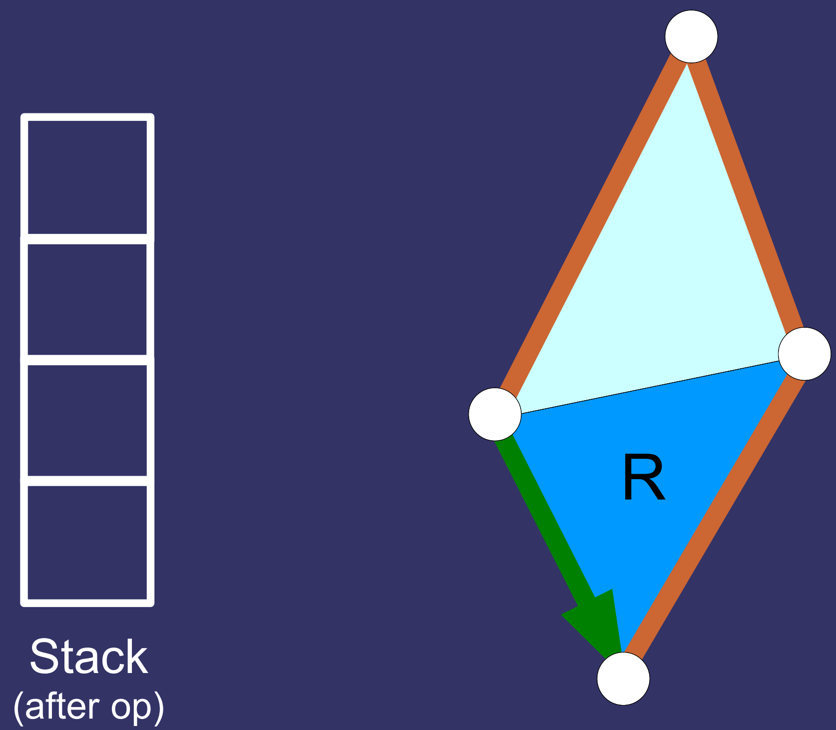
Spirale Reversi in action





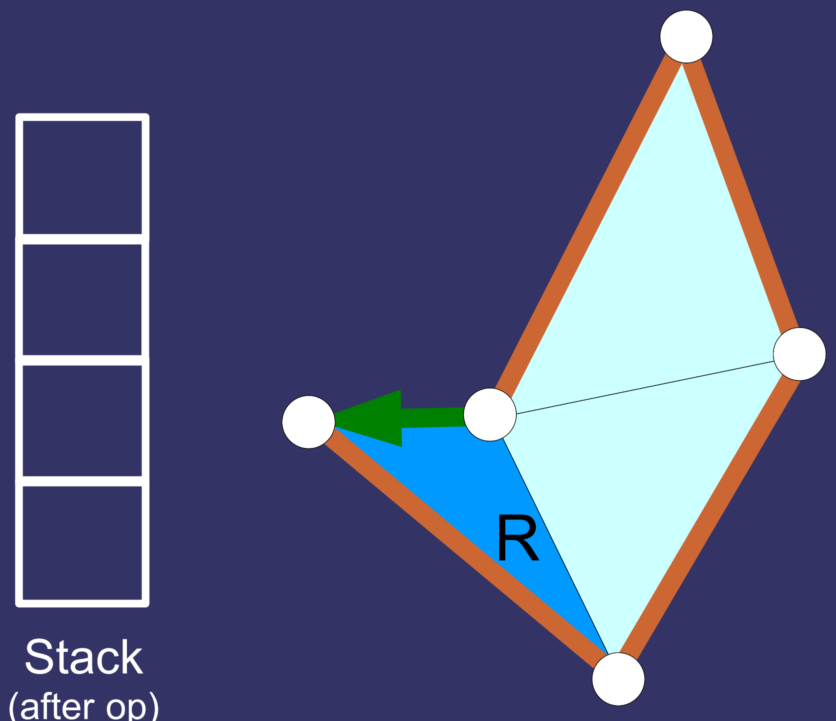
History = CCRRRSLELCRRCRRRE

Spirale Reversi in action



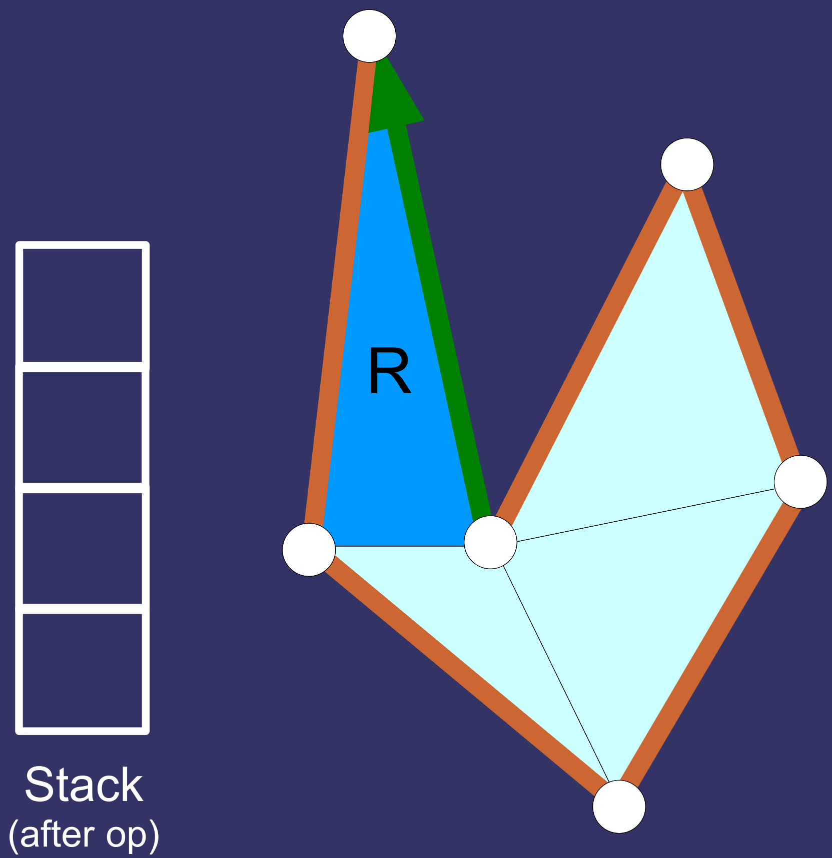
History = CCRRRSLELCRRCRRR

Spirale Reversi in action



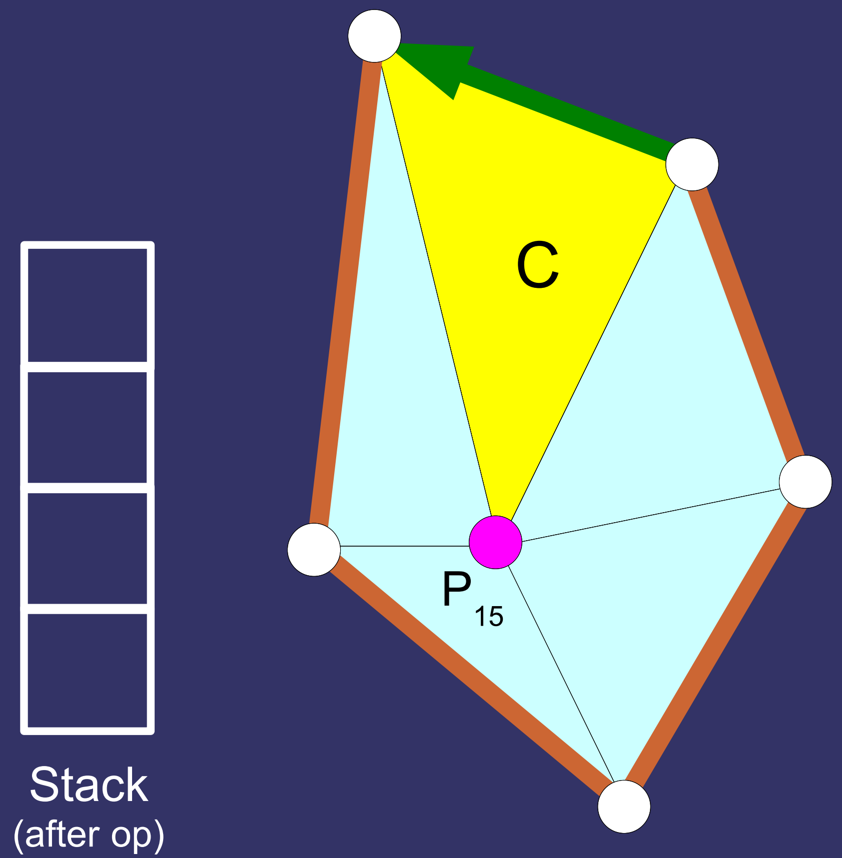
History = CCRRRSLELCRRCRR

Spirale Reversi in action



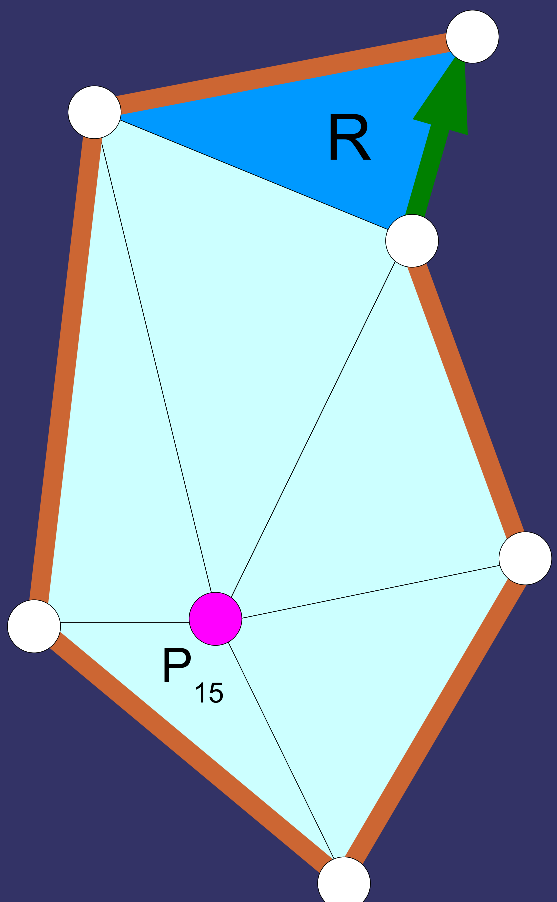
History = CCRRRSLELCRRCR

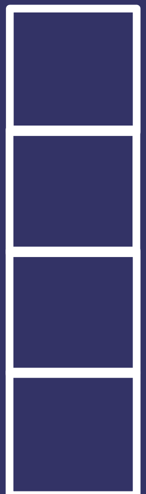
Spirale Reversi in action



History = CCRRRSLELCRRC

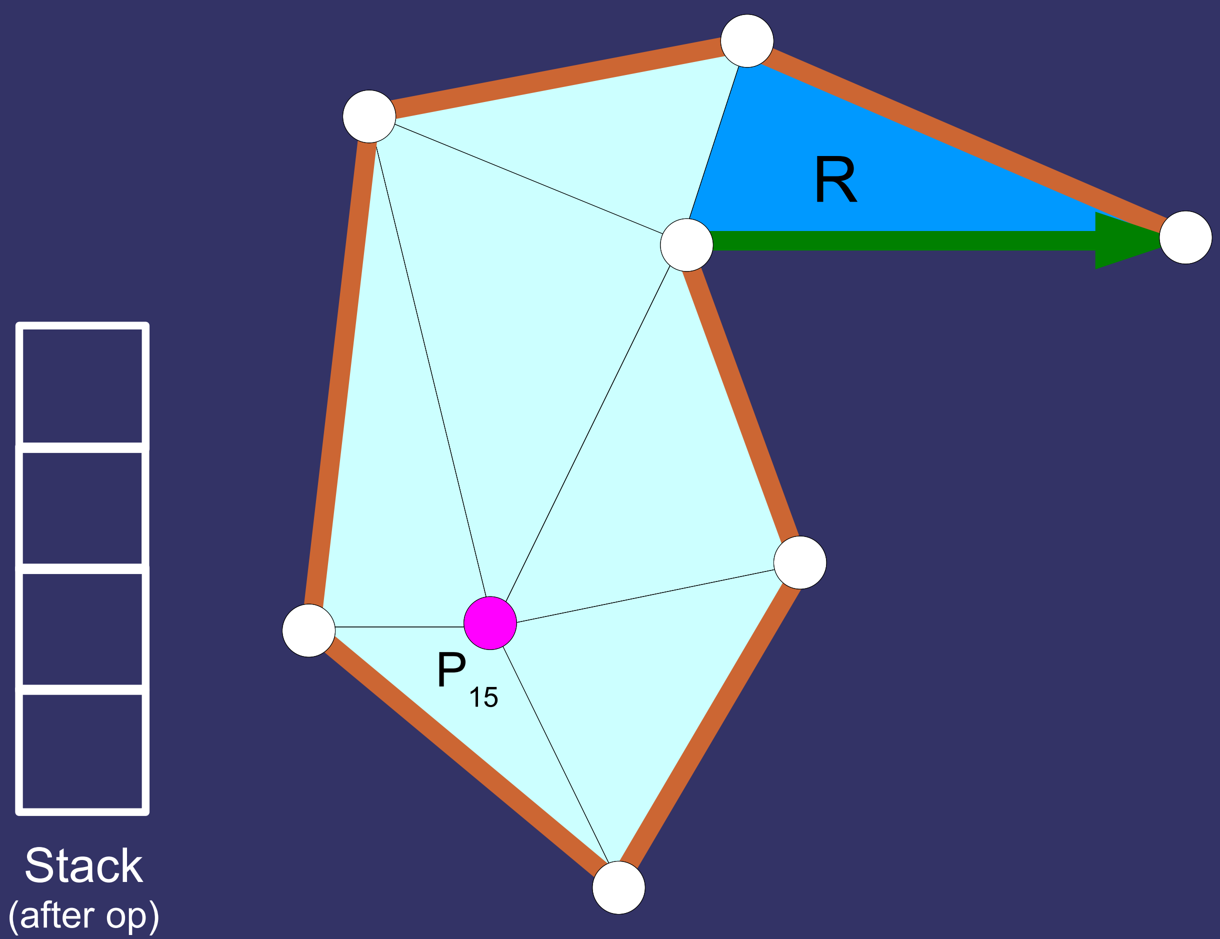
Spirale Reversi in action





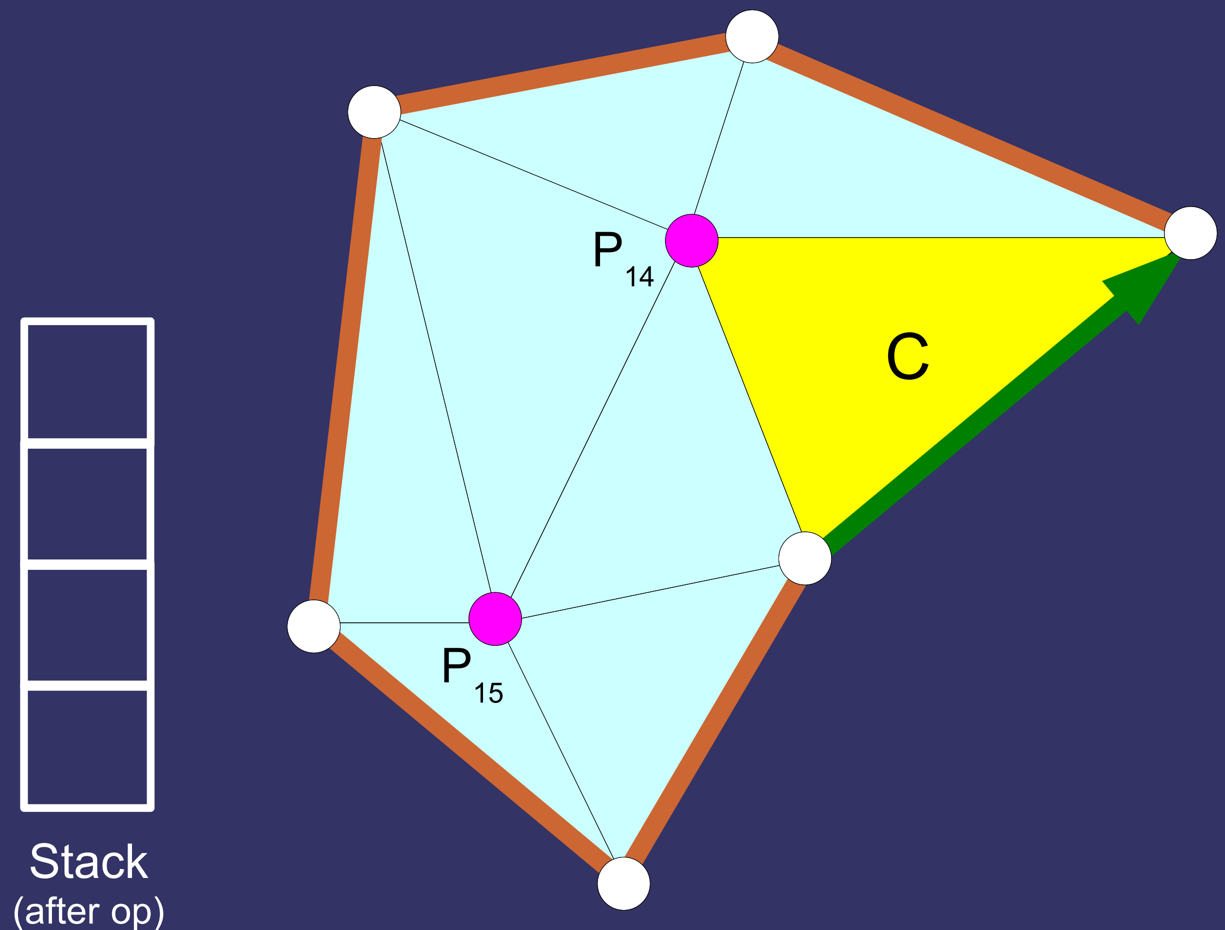
History = CCRRRSLELCRR

Spirale Reversi in action



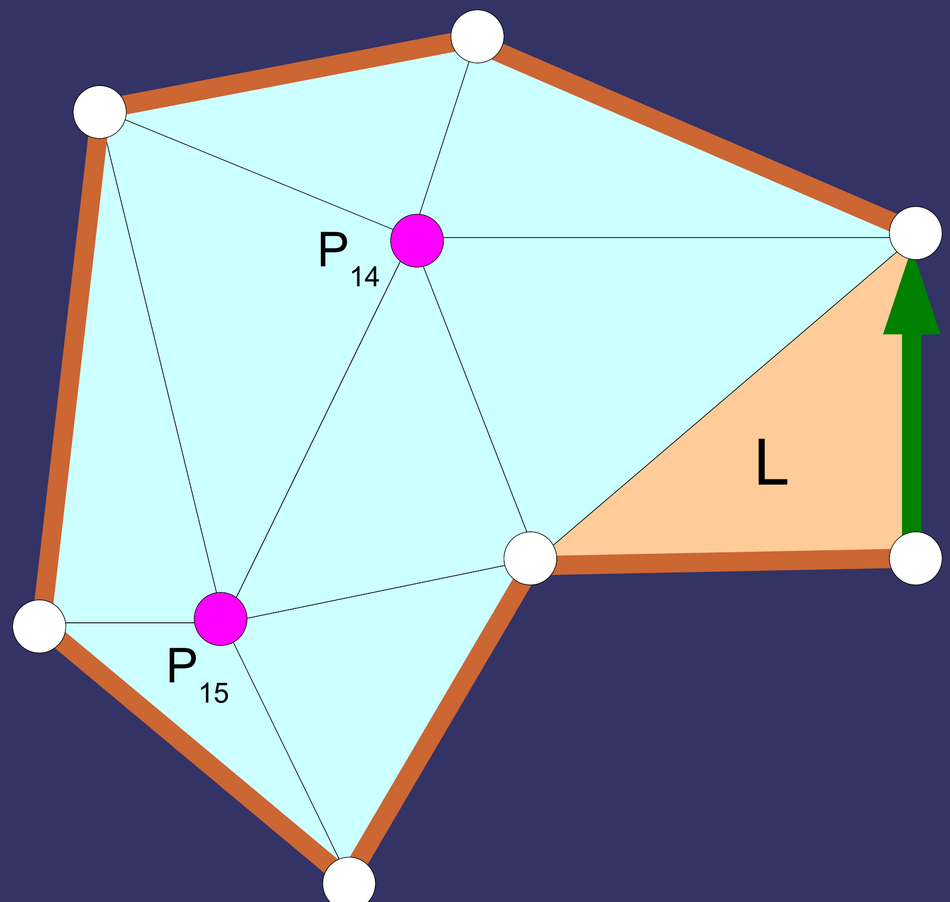
History = CCRRRSLELCR

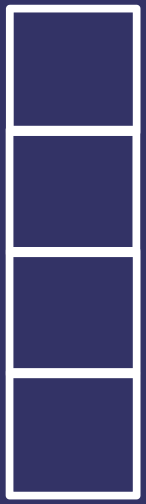
Spirale Reversi in action



History = CCRRRSLELC

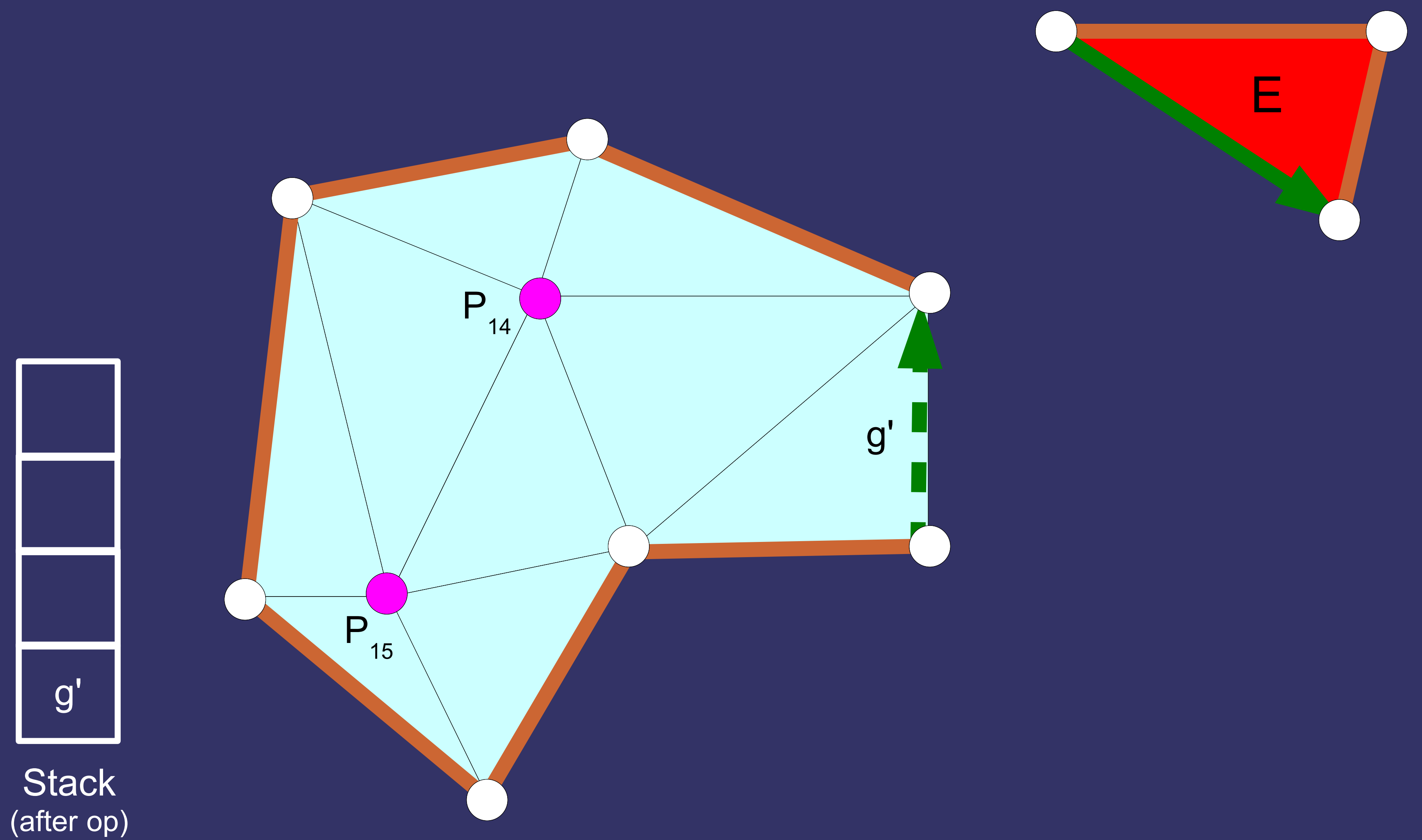
Spirale Reversi in action





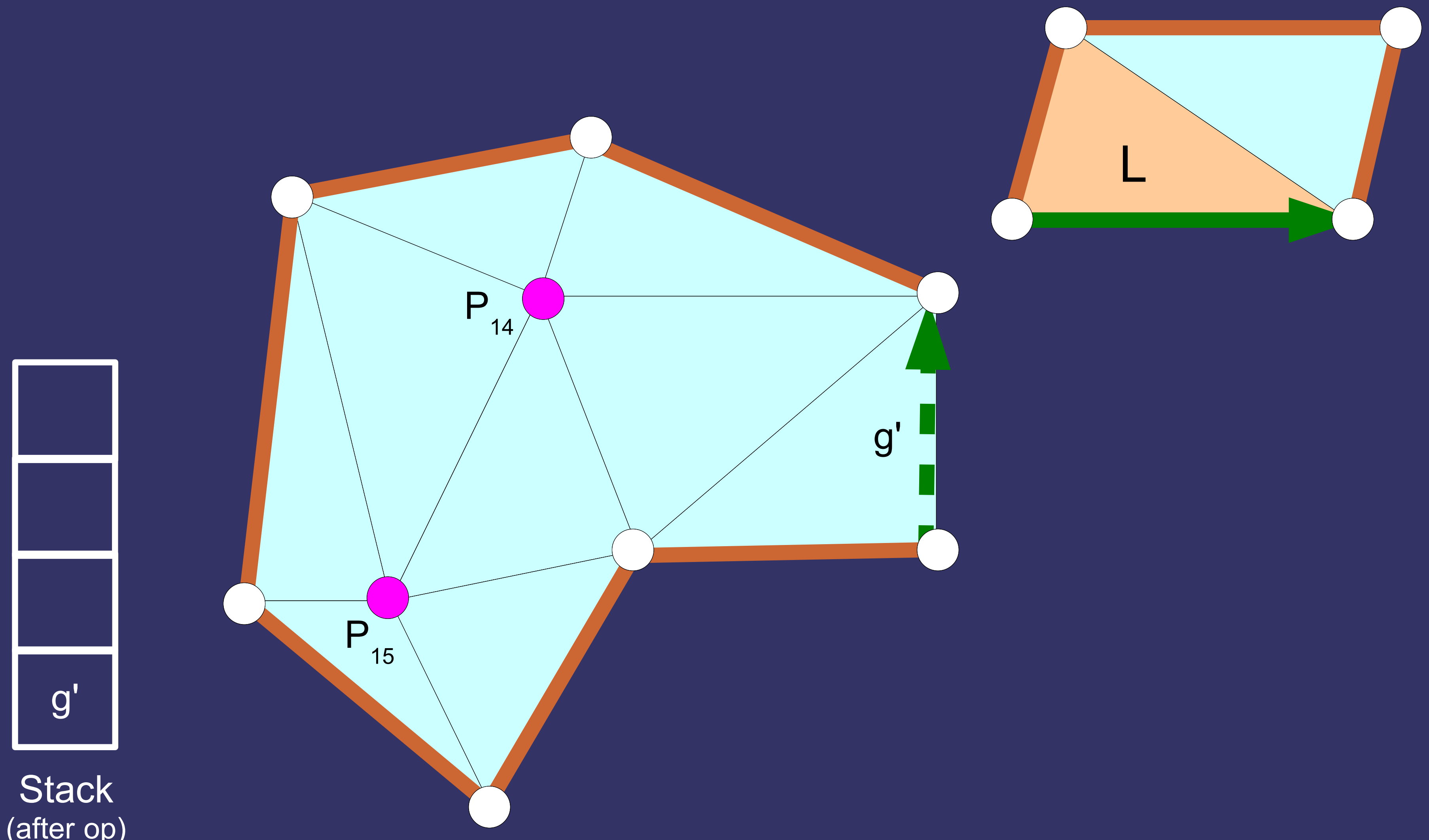
History = CCRRRSLELL

Spirale Reversi in action



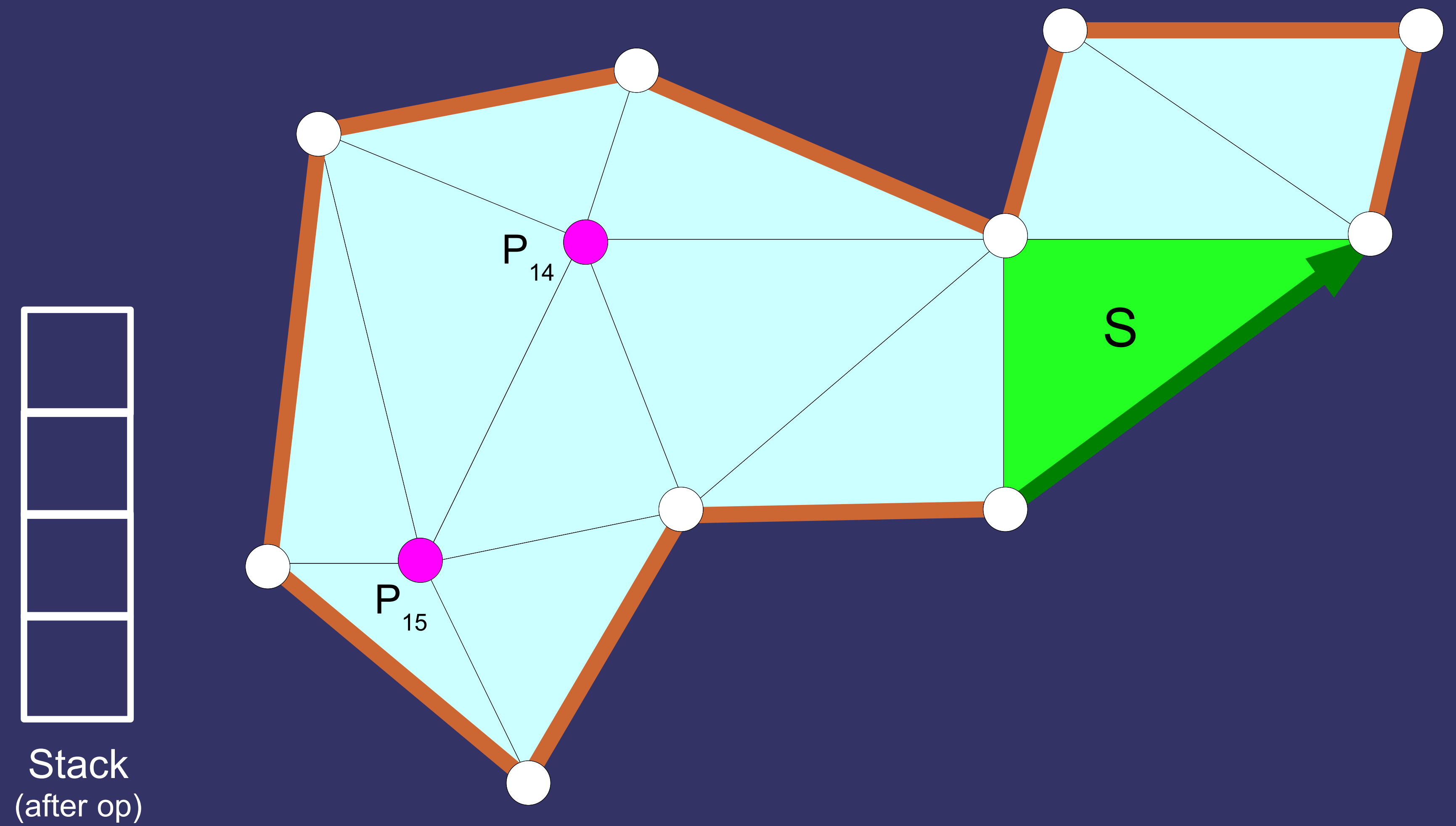
History = CCRRRSLE

Spirale Reversi in action



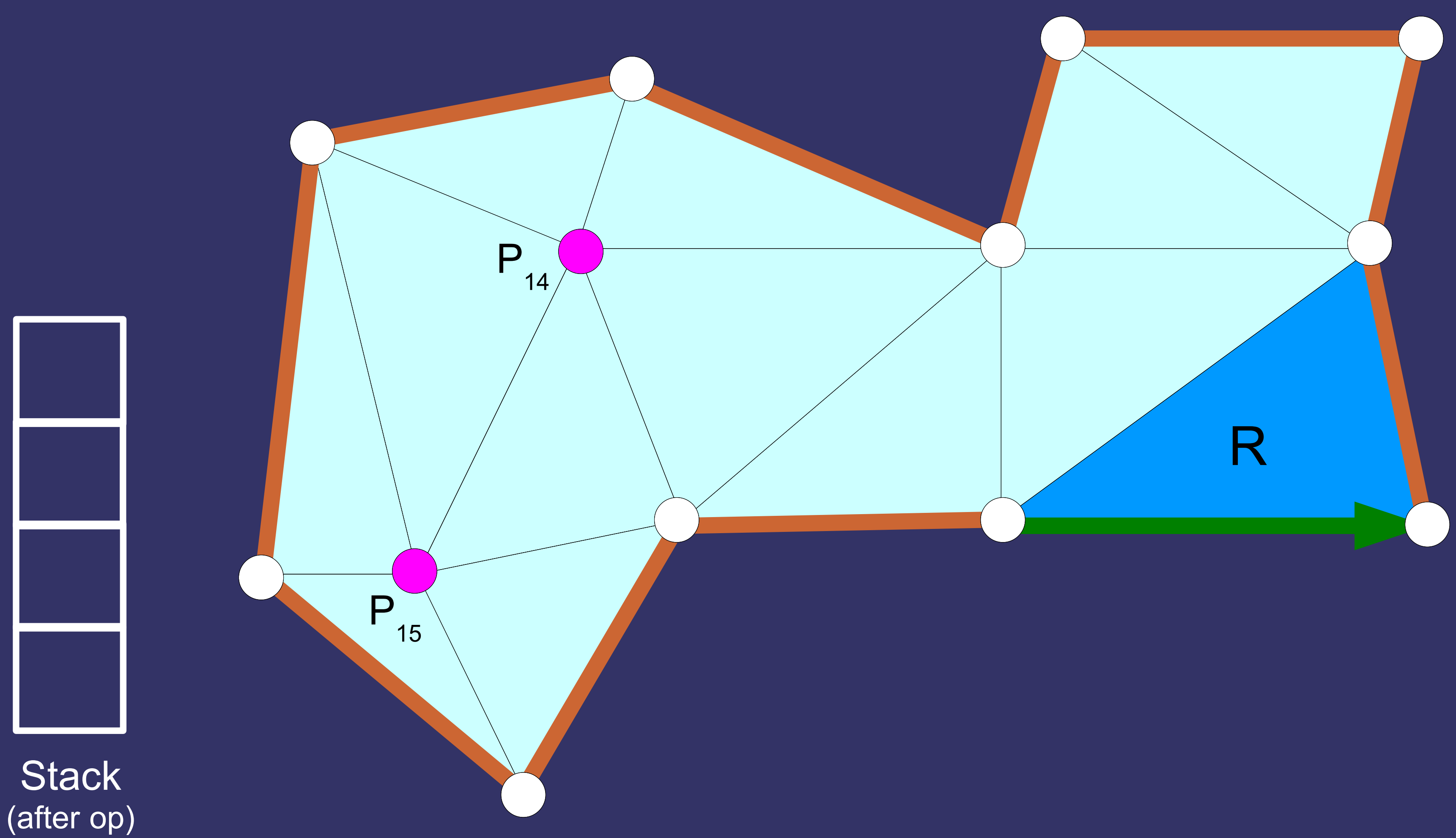
History = CCRRRSL

Spirale Reversi in action



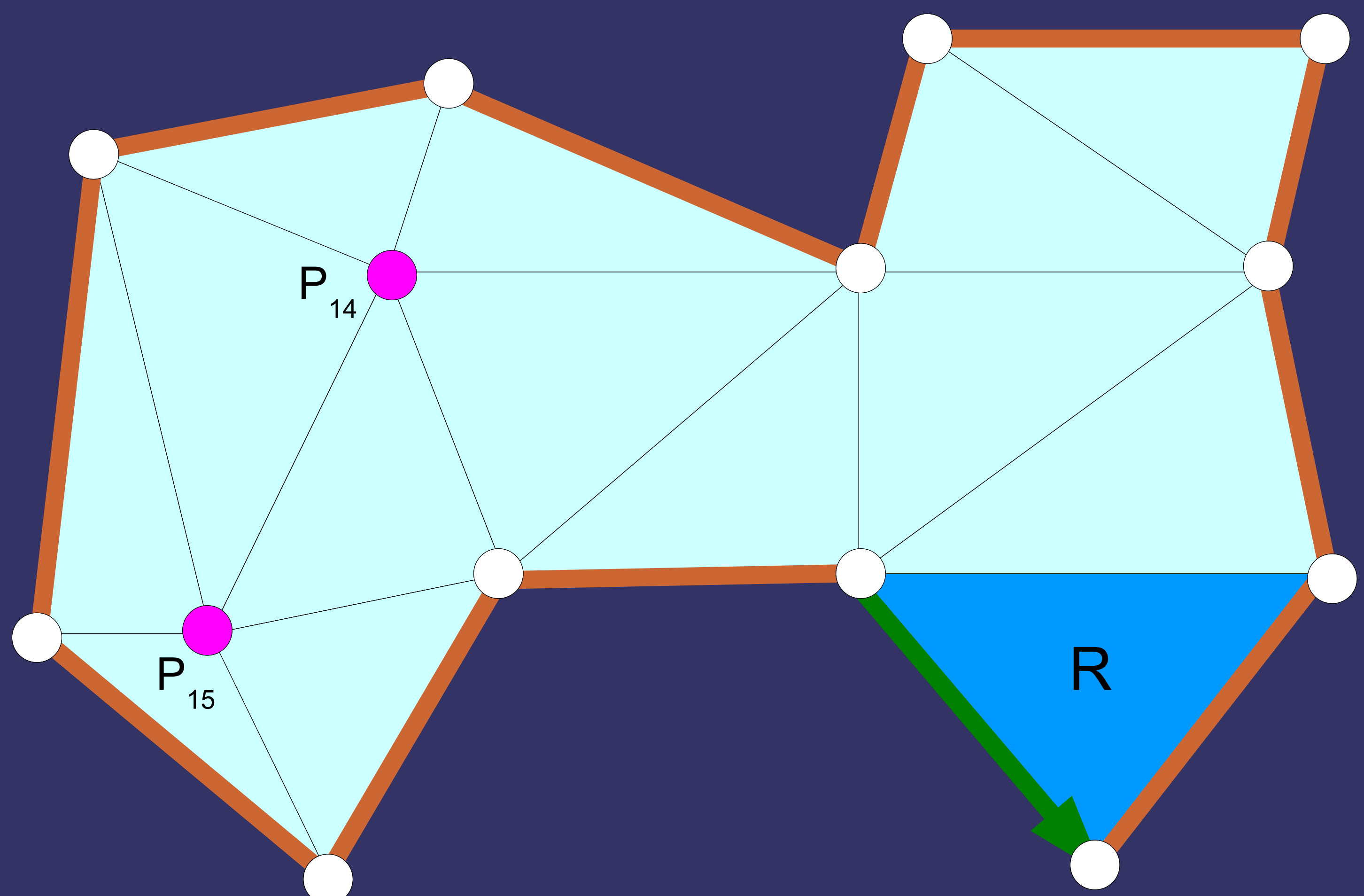
History = CCRRRS

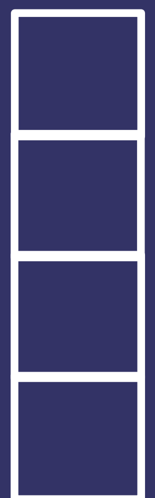
Spirale Reversi in action



History = CCRRR

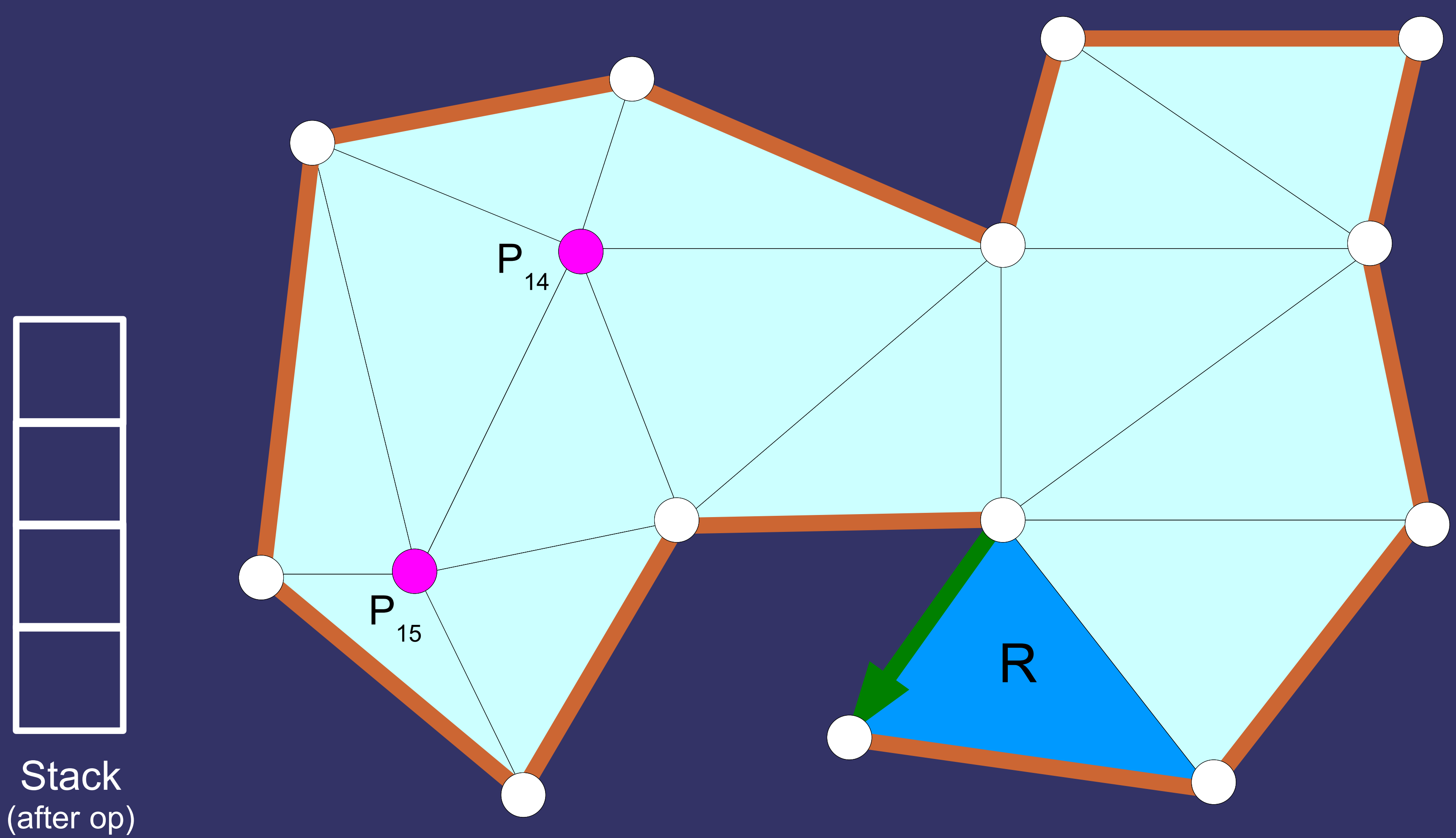
Spirale Reversi in action





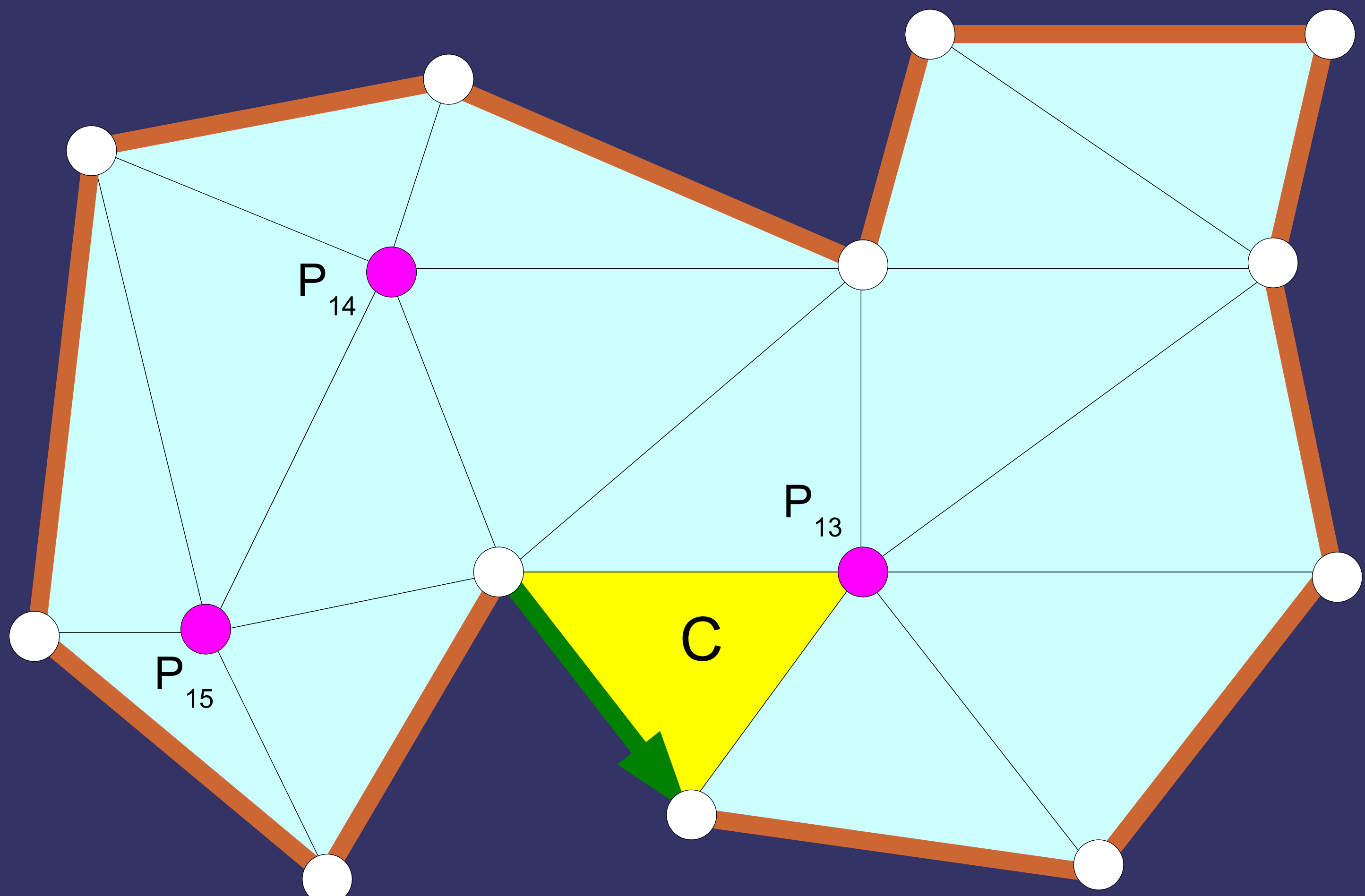
History = CCRR

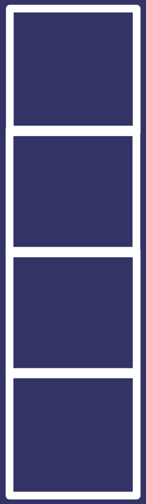
Spirale Reversi in action



History = CCR

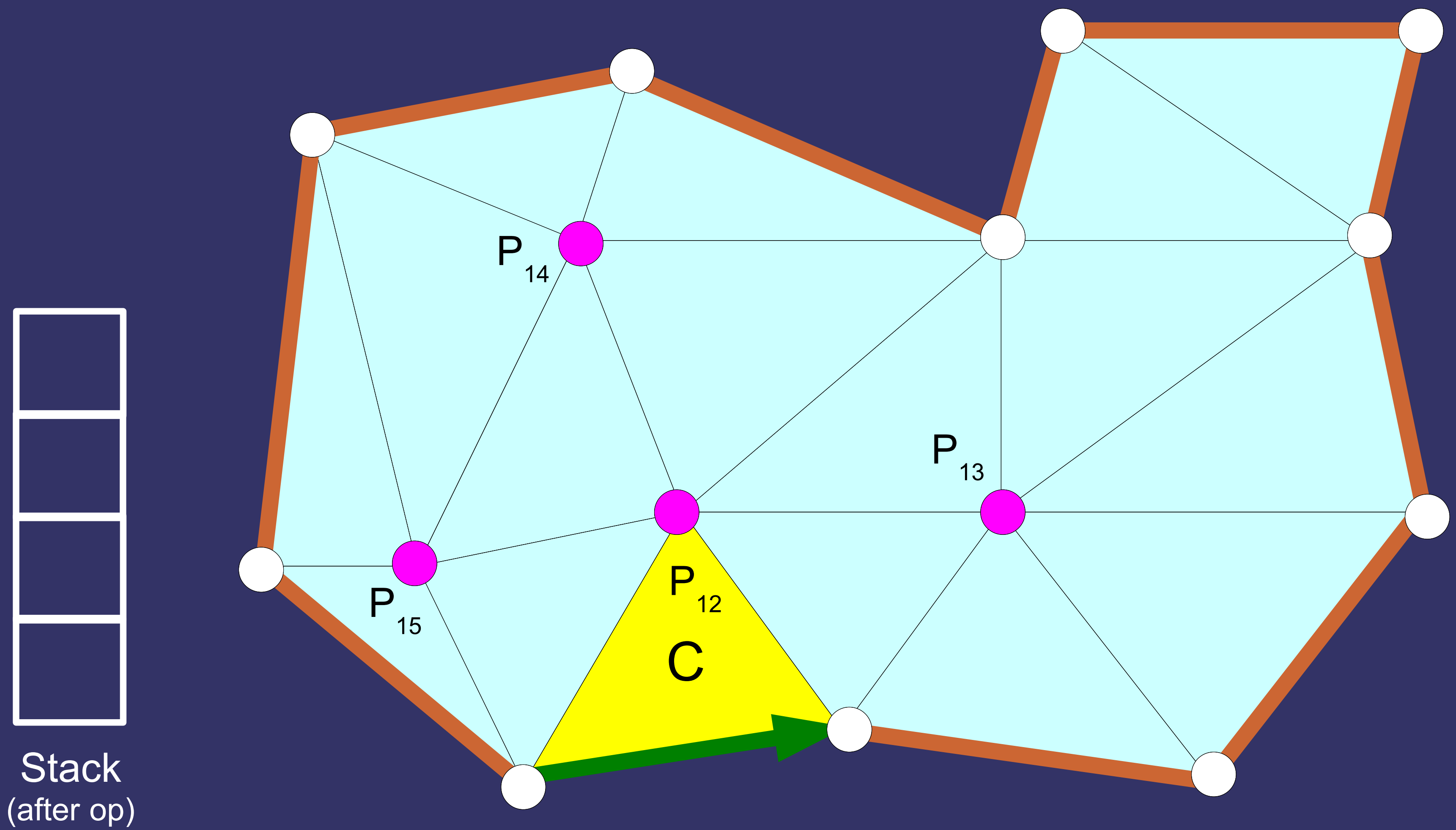
Spirale Reversi in action





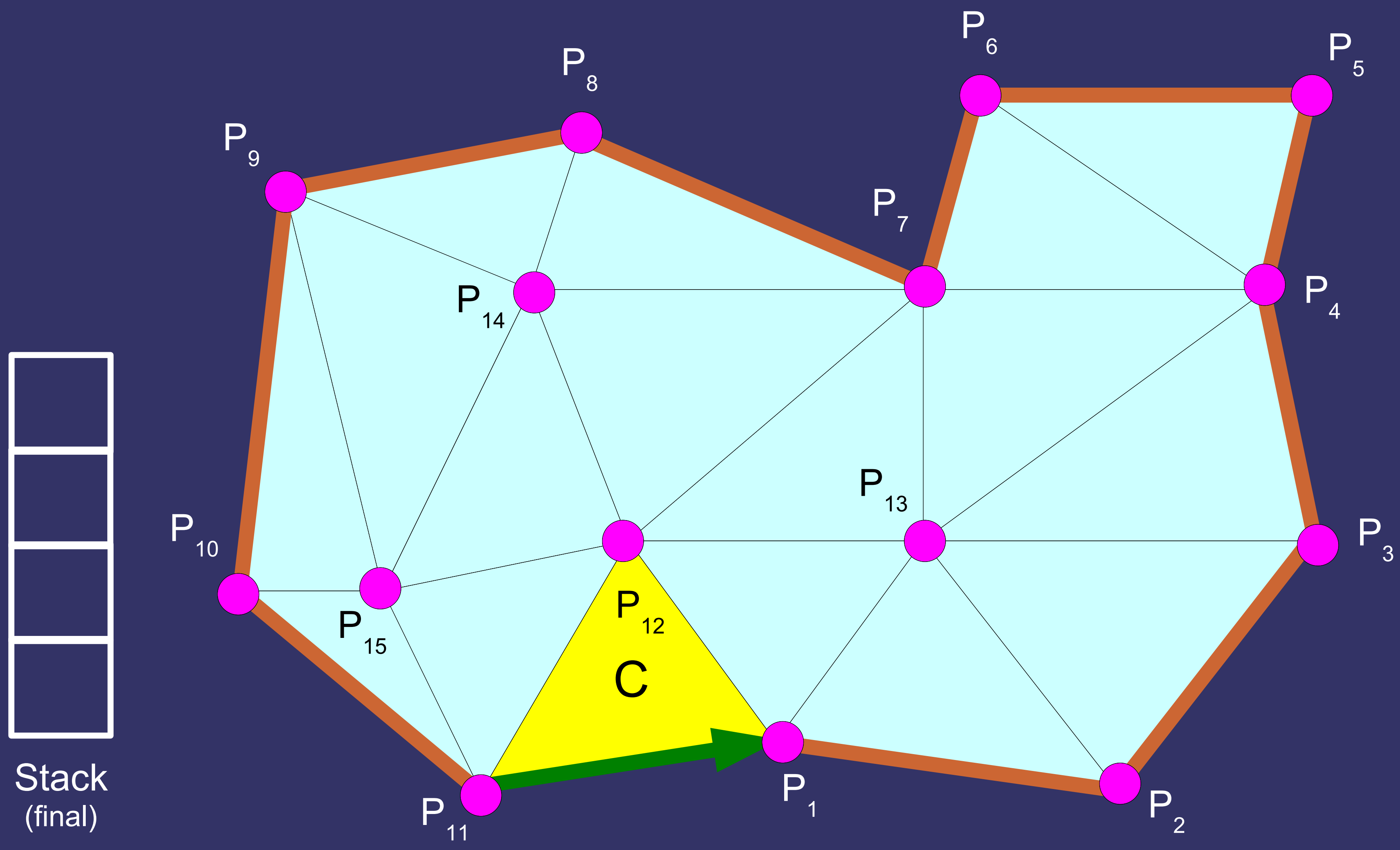
History = CC

Spirale Reversi in action



History = C

Spirale Reversi in action



History = (null)

Patching holes

- While processing a loop, the third vertex v might lie on the boundary of a hole.
- Solution: Introduce M operation which merges the current loop with the hole.

Handling handles

- While processing one loop, we might run into situation where the third vertex v is on some *other* loop created by a previous split operation.
- Solution:
  - Modify split operation S to mark and push additional information about the deferred loop.
  - When we reach a vertex marked as above, execute an M' operation, which merges the two loops.

Discussion

- What about huge meshes?
  - [Isenburg-Gumhold '03] "*Out-of-Core Compression for Gigantic Polygon Meshes*", SIGGRAPH '03.
- Can a different decimation order yield progressively decodable meshes?