logo.png

title.jpg

Polygons

Adam Klepáč

September 18, 2023

Contents

Cryptography on Regular Polygons

Chaining Symmetries

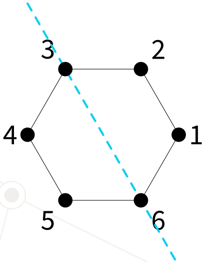
Given two symmetries, s_1 and s_2 of a regular polygon, one can apply them one after the other ('compose' them, like functions).

Chaining Symmetries

Given two symmetries, s_1 and s_2 of a regular polygon, one can apply them one after the other (‘compose’ them, like functions).

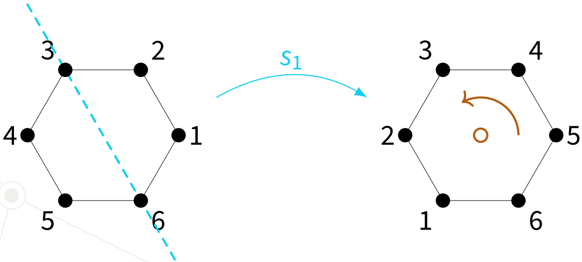
We’ll denote this composition simply by $s_1 s_2$.

Chaining Symmetries – Example



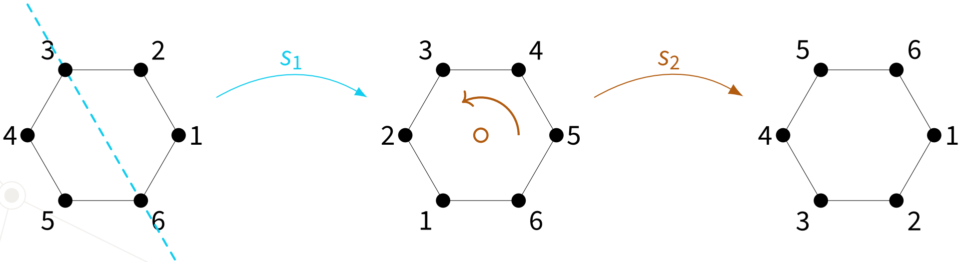
Example of chains of symmetries.

Chaining Symmetries – Example



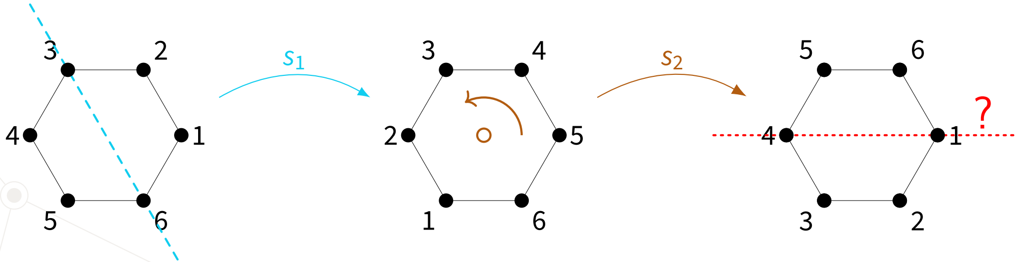
Example of chains of symmetries.

Chaining Symmetries – Example



Example of chains of symmetries.

Chaining Symmetries – Example



Example of chains of symmetries.

Chaining Symmetries – How Many Do We Need?

Discounting point symmetry, an n-gon has $2n$ symmetries.

Chaining Symmetries – How Many Do We Need?

Discounting point symmetry, an n-gon has $2n$ symmetries.

Two symmetries can ‘combine’ to create a different symmetry.

Chaining Symmetries – How Many Do We Need?

Discounting point symmetry, an n-gon has $2n$ symmetries.

Two symmetries can ‘combine’ to create a different symmetry.

Natural question: How many (and which) symmetries of a regular polygon do I need to get all the others?

Chaining Symmetries – How Many Do We Need?

Discounting point symmetry, an n-gon has $2n$ symmetries.

Two symmetries can 'combine' to create a different symmetry.

Natural question: How many (and which) symmetries of a regular polygon do I need to get all the others?

For example,

- if s_1 is any line symmetry and s_2 is a rotation by 60° counter-clockwise, then $s_2^3 s_1$ (s_2^3 means $s_2 s_2 s_2$) reflects a hexagon through a line perpendicular to the line of s_1.

Chaining Symmetries – How Many Do We Need?

Discounting point symmetry, an n-gon has $2n$ symmetries.

Two symmetries can 'combine' to create a different symmetry.

Natural question: How many (and which) symmetries of a regular polygon do I need to get all the others?

For example,

- if s_1 is any line symmetry and s_2 is a rotation by 60° counter-clockwise, then $s_2^3 s_1$ (s_2^3 means $s_2 s_2 s_2$) reflects a hexagon through a line perpendicular to the line of s_1.
- if s_1 is a rotation by 120° clockwise and s_2 is a reflection through a vertical line passing through the top vertex, then $s_1 s_2$ is a reflection through the line given by the rotation of the line of s_2 60° clockwise.

Chaining Symmetries – How Many Do We Need?

Actually, for a general n-gon, we need only **two**:

Chaining Symmetries – How Many Do We Need?

Actually, for a general n-gon, we need only **two**:

- rotation by $360^\circ/n$ in any direction (we'll denote it r),

Chaining Symmetries – How Many Do We Need?

Actually, for a general n-gon, we need only **two**:

- rotation by $360^\circ/n$ in any direction (we'll denote it r),
- any reflection (we'll denote it s).

Chaining Symmetries – Triangle

Let r be the rotation by 120° and s any line symmetry.

Chaining Symmetries – Triangle

Let r be the rotation by 120° and s any line symmetry.

- The other two rotational symmetries are r^2 and r^3.

Chaining Symmetries – Triangle

Let r be the rotation by 120° and s any line symmetry.

- The other two rotational symmetries are r^2 and r^3.
- The other two line symmetries are rs and r^2s.

Chaining Symmetries – Triangle

Let r be the rotation by 120° and s any line symmetry.

- The other two rotational symmetries are r^2 and r^3.
- The other two line symmetries are rs and r^2s.
- Therefore, all the symmetries of an equilateral triangle are

$$\{r, r^2, r^3, s, rs, r^2s\}.$$

Chaining Symmetries – General Algorithm

In general, to create all symmetries, one needs a rotation by an angle $k \cdot 360^\circ / n$ where k **doesn't share a prime factor** with n (in other words, the fraction $\frac{k}{n}$ cannot be simplified) and any one line symmetry.

Chaining Symmetries – General Algorithm

In general, to create all symmetries, one needs a rotation by an angle $k \cdot 360^\circ/n$ where k **doesn't share a prime factor** with n (in other words, the fraction $\frac{k}{n}$ cannot be simplified) and any one line symmetry.

Why?

- If k shares factors with n, then you can never get rotation by $360^\circ/n$.

Chaining Symmetries – General Algorithm

In general, to create all symmetries, one needs a rotation by an angle $k \cdot 360^\circ/n$ where k **doesn't share a prime factor** with n (in other words, the fraction $\frac{k}{n}$ cannot be simplified) and any one line symmetry.

Why?

- If k shares factors with n, then you can never get rotation by $360^\circ/n$.
- Two symmetries can never produce a rotation.

Chaining Symmetries – General Algorithm

In general, to create all symmetries, one needs a rotation by an angle $k \cdot 360^\circ/n$ where k **doesn't share a prime factor** with n (in other words, the fraction $\frac{k}{n}$ cannot be simplified) and any one line symmetry.

Why?

- If k shares factors with n, then you can never get rotation by $360^\circ/n$.
- Two symmetries can never produce a rotation.
- Two rotations can never produce a symmetry.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360^\circ/n$ such that k doesn't share factors with n and a line symmetry s.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360^\circ / n$ such that k doesn't share factors with n and a line symmetry s.

If you need to calculate a rotation, then

1. First measure the angle **counter-clockwise**.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360^\circ / n$ such that k doesn't share factors with n and a line symmetry s.

If you need to calculate a rotation, then

1. First measure the angle **counter-clockwise**.
2. Find a such that r^a is the rotation by $360^\circ / n$.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360^\circ/n$ such that k doesn't share factors with n and a line symmetry s.

If you need to calculate a rotation, then

1. First measure the angle **counter-clockwise**.
2. Find a such that r^a is the rotation by $360^\circ/n$.
3. Then, find b such that $(r^a)^b = r^{ab}$ is your desired rotation.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360^\circ/n$ such that k doesn't share factors with n and a line symmetry s.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360^\circ/n$ such that k doesn't share factors with n and a line symmetry s.

If you need to calculate a reflection, then

1. Find a such that r^a is the rotation by $360^\circ/n$.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360^\circ/n$ such that k doesn't share factors with n and a line symmetry s.

If you need to calculate a reflection, then

1. Find a such that r^a is the rotation by $360^\circ/n$.
2. Determine the angle **in any direction** between your given line of symmetry s and the reflection you want.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360^\circ / n$ such that k doesn't share factors with n and a line symmetry s.

If you need to calculate a reflection, then

1. Find a such that r^a is the rotation by $360^\circ / n$.
2. Determine the angle **in any direction** between your given line of symmetry s and the reflection you want.
3. Find b such that r^{ab} is a rotation **in the opposite direction** by **twice** the angle from the previous step.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360° / n$ such that k doesn't share factors with n and a line symmetry s.

If you need to calculate a reflection, then

1. Find a such that r^a is the rotation by $360° / n$.
2. Determine the angle **in any direction** between your given line of symmetry s and the reflection you want.
3. Find b such that r^{ab} is a rotation **in the opposite direction** by **twice** the angle from the previous step.
4. $r^{ab}s$ is your desired reflection.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360^\circ/n$ such that k doesn't share factors with n and a line symmetry s.

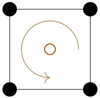
If you need to calculate a reflection, then

1. Find a such that r^a is the rotation by $360^\circ/n$.
2. Determine the angle **in any direction** between your given line of symmetry s and the reflection you want.
3. Find b such that r^{ab} is a rotation **in the opposite direction** by **twice** the angle from the previous step.
4. $r^{ab}s$ is your desired reflection.

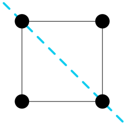
Voluntary HW: *Why* does this algorithm work?

Chaining Symmetries – Algorithm Example

We're given two symmetries of the square:

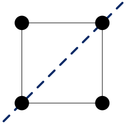


rotation r by 270°



reflection s over the first diagonal

and want to produce



reflection over the second diagonal

Chaining Symmetries – Algorithm Example

We're given two symmetries of the square: rotation r by 270° counter-clockwise and reflection s over the first diagonal.

How to produce reflection over the other diagonal?

Chaining Symmetries – Algorithm Example

We're given two symmetries of the square: rotation r by 270° counter-clockwise and reflection s over the first diagonal.

How to produce reflection over the other diagonal?

We use the algorithm.

1. Repeating r three times gives the rotation by 90° counter-clockwise, that is, $a = 3$.

Chaining Symmetries – Algorithm Example

We're given two symmetries of the square: rotation r by 270° counter-clockwise and reflection s over the first diagonal.

How to produce reflection over the other diagonal?

We use the algorithm.

1. Repeating r three times gives the rotation by 90° counter-clockwise, that is, $a = 3$.
2. The angle between the two diagonals is 90° in any direction.

Chaining Symmetries – Algorithm Example

We're given two symmetries of the square: rotation r by 270° counter-clockwise and reflection s over the first diagonal.

How to produce reflection over the other diagonal?

We use the algorithm.

1. Repeating r three times gives the rotation by 90° counter-clockwise, that is, $a = 3$.
2. The angle between the two diagonals is 90° in any direction.
3. Repeating the rotation from step 1 two times (that is, $b = 2$) and then using s gives the desired symmetry – in this case it's $(r^3)^2 s = r^6 s$. Of course, r^4 is rotation by 360° which does nothing, so the final symmetry is $r^2 s$.