

logo.png

title.jpg

# POLYGONS

Adam Klepáč

October 1, 2023

# CONTENTS



## General Polygons

### Encoding Messages Using Symmetries

# GENERAL POLYGONS

The background of the slide features three large, overlapping triangles. A yellow triangle is on the left, a cyan triangle is on the right, and a green triangle is at the bottom center, partially overlapping the other two.

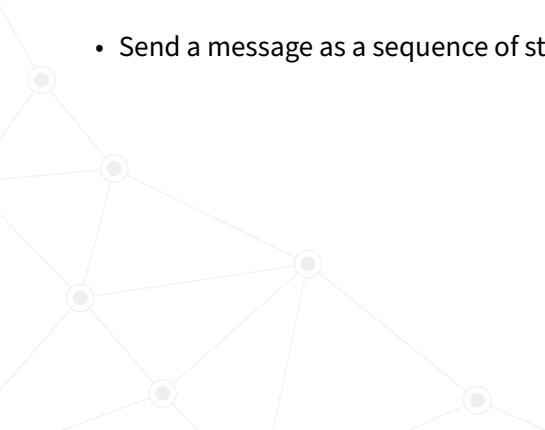
1

## ENCODING MESSAGES USING SYMMETRIES



# ENCODING MESSAGES – GENERAL IDEA

- Send a message as a sequence of strings which **can be decoded on the other side**.



# ENCODING MESSAGES – GENERAL IDEA

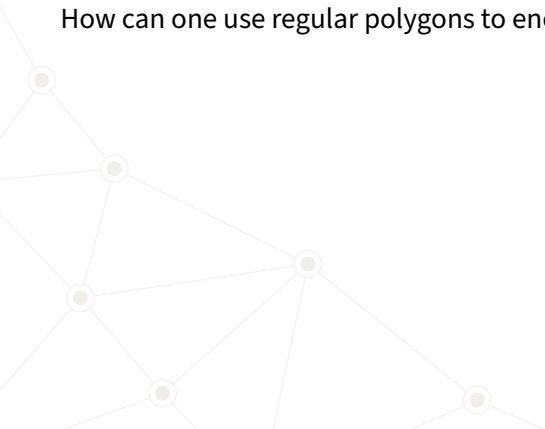
- Send a message as a sequence of strings which **can be decoded on the other side**.
- The decoding should require a 'key' which is **statistically impossible to determine quickly**.

# ENCODING MESSAGES – GENERAL IDEA

- Send a message as a sequence of strings which **can be decoded on the other side**.
- The decoding should require a 'key' which is **statistically impossible to determine quickly**.
- The encoding and decoding must be done procedurally – requires a system with concrete rules and a **limited** (but huge) **number of combinations**.

# ENCODING MESSAGES – REGULAR POLYGONS

How can one use regular polygons to encode messages?

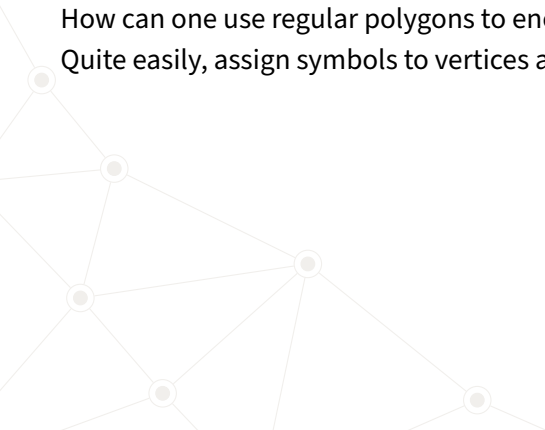




# ENCODING MESSAGES – REGULAR POLYGONS

How can one use regular polygons to encode messages?

Quite easily, assign symbols to vertices and choose one **main** vertex.

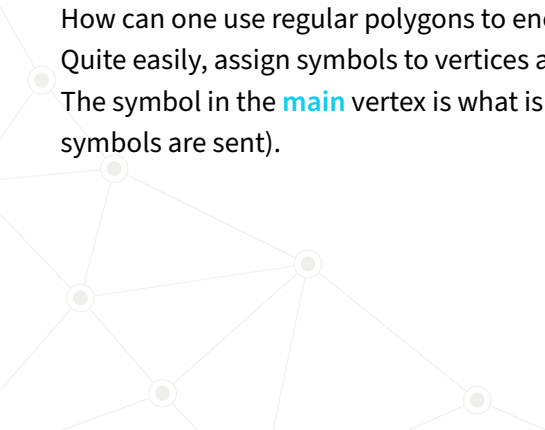


# ENCODING MESSAGES – REGULAR POLYGONS

How can one use regular polygons to encode messages?

Quite easily, assign symbols to vertices and choose one **main** vertex.

The symbol in the **main** vertex is what is sent during each tick (time interval during which symbols are sent).



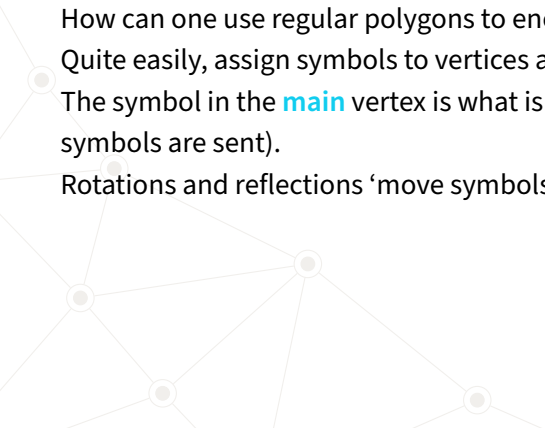
# ENCODING MESSAGES – REGULAR POLYGONS

How can one use regular polygons to encode messages?

Quite easily, assign symbols to vertices and choose one **main** vertex.

The symbol in the **main** vertex is what is sent during each tick (time interval during which symbols are sent).

Rotations and reflections ‘move symbols between vertices’.



# ENCODING MESSAGES – REGULAR POLYGONS

How can one use regular polygons to encode messages?

Quite easily, assign symbols to vertices and choose one **main** vertex.

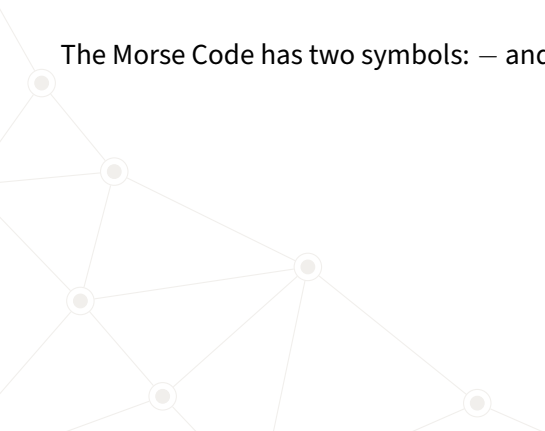
The symbol in the **main** vertex is what is sent during each tick (time interval during which symbols are sent).

Rotations and reflections ‘move symbols between vertices’.

This means that after applying a rotation or reflection, a (in most cases) different symbol will appear in the main vertex.

# ENCODING MESSAGES – MORSE CODE

The Morse Code has two symbols: — and ·.

A decorative graphic in the bottom-left corner of the slide. It consists of several light gray lines connecting small, light gray circular nodes. The nodes are arranged in a way that forms a series of interconnected triangles and polygons, creating a network-like or geometric pattern. The lines are thin and the nodes are small, giving it a subtle, modern look.

# ENCODING MESSAGES – MORSE CODE

The Morse Code has two symbols: — and ·.

We also need to be able to determine spaces between words – we can assign one vertex a ‘blank’, meaning space.

# ENCODING MESSAGES – MORSE CODE

The Morse Code has two symbols: — and ·.

We also need to be able to determine spaces between words – we can assign one vertex a ‘blank’, meaning space.

This means that to send messages written Morse Code using regular polygons, we need three vertices – a triangle.

# ENCODING MESSAGES – MORSE CODE

Let the top vertex be main and let's choose a rotation  $r = \curvearrowright 120^\circ$  and a reflection  $s$  depicted below.

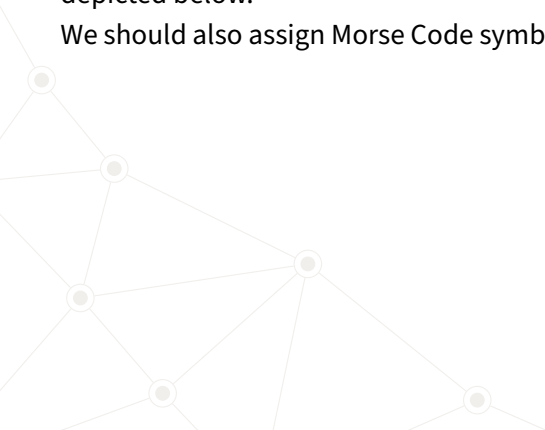




# ENCODING MESSAGES – MORSE CODE

Let the top vertex be main and let's choose a rotation  $r = \curvearrowright 120^\circ$  and a reflection  $s$  depicted below.

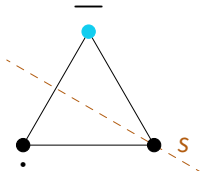
We should also assign Morse Code symbols to two of the vertices. We can do so randomly.



# ENCODING MESSAGES – MORSE CODE

Let the top vertex be main and let's choose a rotation  $r = \curvearrowright 120^\circ$  and a reflection  $s$  depicted below.

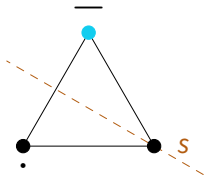
We should also assign Morse Code symbols to two of the vertices. We can do so randomly.



## ENCODING MESSAGES – MORSE CODE

Let the top vertex be main and let's choose a rotation  $r = \curvearrowright 120^\circ$  and a reflection  $s$  depicted below.

We should also assign Morse Code symbols to two of the vertices. We can do so randomly.



The **top vertex** is the main one, which means that the symbol above it will get sent.

# ENCODING MESSAGES – MORSE CODE

The important property of symmetries is that two different sequences can give the same symmetry.



# ENCODING MESSAGES – MORSE CODE

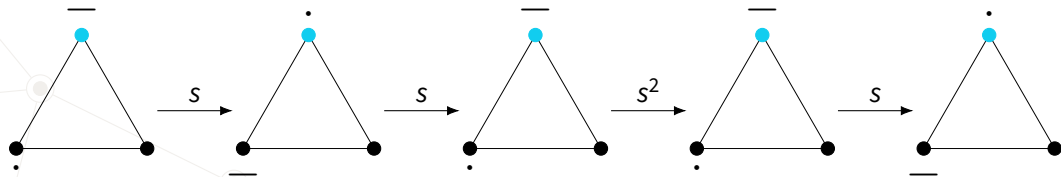
The important property of symmetries is that two different sequences can give the same symmetry. Meaning, we can encode one message **MANY** different ways.



## ENCODING MESSAGES – MORSE CODE

The important property of symmetries is that two different sequences can give the same symmetry. Meaning, we can encode one message **MANY** different ways.

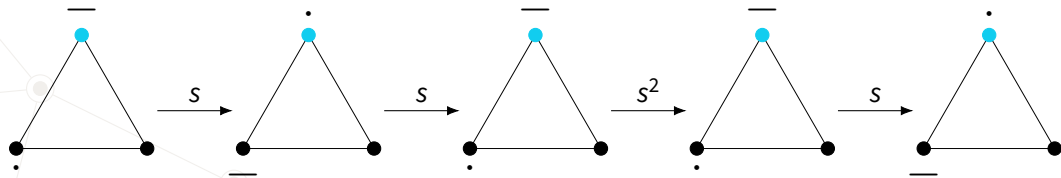
For example, we can send the letter P ( $\cdot - - \cdot$ ) like this:



## ENCODING MESSAGES – MORSE CODE

The important property of symmetries is that two different sequences can give the same symmetry. Meaning, we can encode one message **MANY** different ways.

For example, we can send the letter P ( $\cdot - - \cdot$ ) like this:

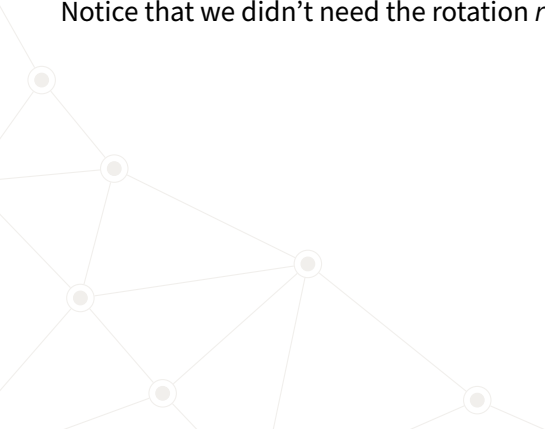


Or, as a sequence

$S, S, S^2, S.$

# ENCODING MESSAGES – MORSE CODE

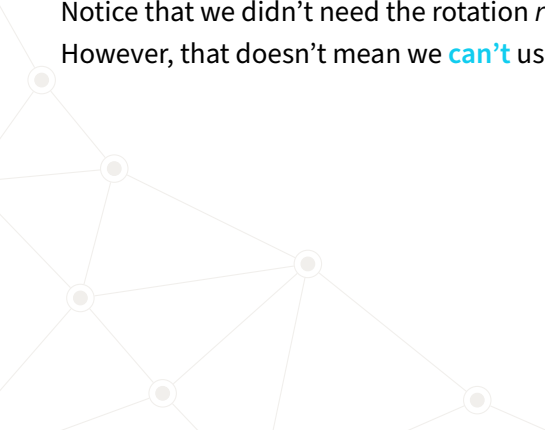
Notice that we didn't need the rotation  $r$  at all for this example.





# ENCODING MESSAGES – MORSE CODE

Notice that we didn't need the rotation  $r$  at all for this example.  
However, that doesn't mean we **can't** use it.



# ENCODING MESSAGES – MORSE CODE

Notice that we didn't need the rotation  $r$  at all for this example.

However, that doesn't mean we **can't** use it.

Actually, it's often beneficial to encode messages using different combinations of symmetries for security.

# ENCODING MESSAGES – MORSE CODE

Notice that we didn't need the rotation  $r$  at all for this example.

However, that doesn't mean we **can't** use it.

Actually, it's often beneficial to encode messages using different combinations of symmetries for security.

The letter  $\cdot - - \cdot$  can also be sent for example like this

$$r^2, sr, sr^2, rsr.$$