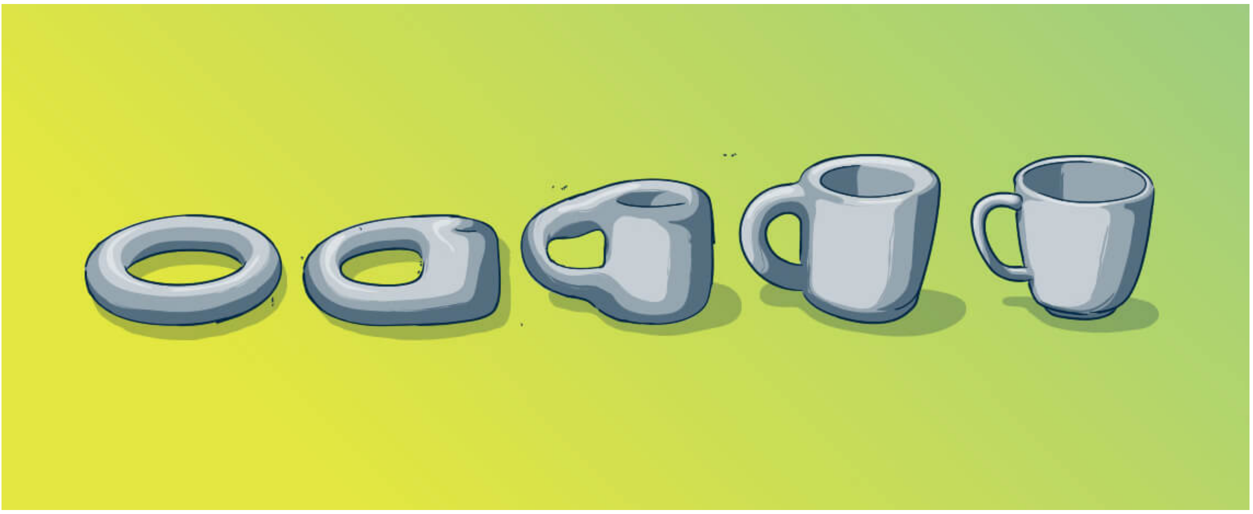
Introduction:

Topology is a branch of mathematics which works with objects that are unaffected by continuous deformation, we will work with 3-d surfaces. The notion of continuous deformation can be simplified to the idea of not “breaking” or “ripping” the surface we are working with. The common joke to make at this point is: “How can you tell you’re talking to a topologist? They can't tell the difference between a donut and a coffee cup”.



As you can see by elongating one side and “indenting” the ring without poking a hole in the bottom you can (theoretically) go from a donut to a coffee cup. The property of two objects being “the same” is referred to as two objects being ***homeomorphic*** to each other, so a donut is homeomorphic to a coffee cup. And this idea brings us to the motivation of the program.

(I)Motivation:

It has been proven that any 3-d object is homeomorphic to either:

1. A set of connected tori(donuts)
2. A set of connected projective planes

(Note: a sphere is considered a set of 0 tori. *Tricky mathematicians.*)

So the ***goal*** of this program will be to take some given object(*word*) and (provided its a surface) reduce it to its basic components of either Tori or Projective Planes.

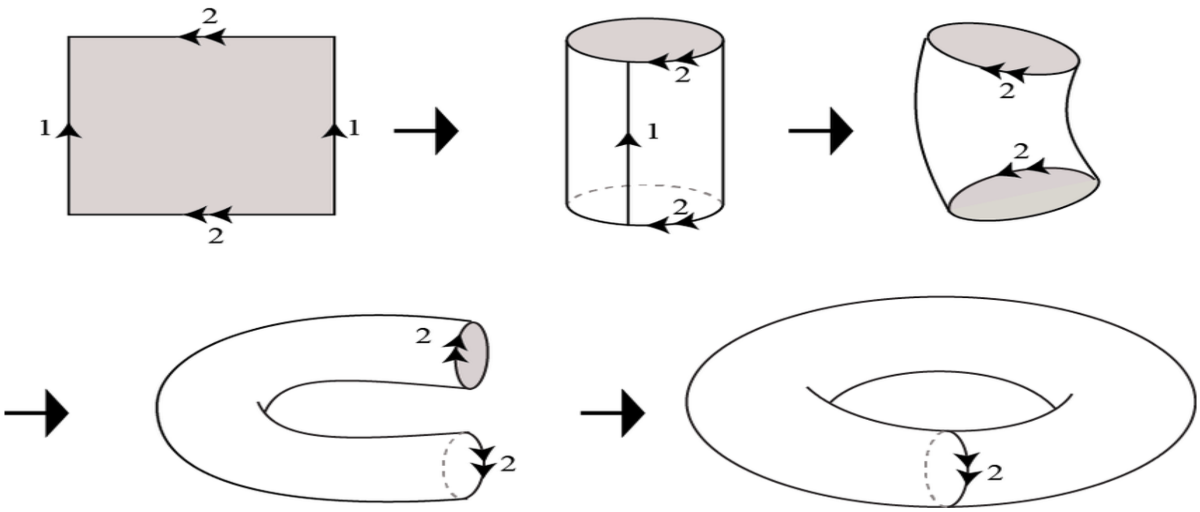
(II)Object Representation:

A. Planar models:

A planar model works by encoding information of higher dimensional objects into the 2-d plane. Each edge has a label and an orientation. To create the object from the planar model you identify (or glue) labels together so that the edge orientations match.

In the below diagram one can see how by combining the edges labeled “1” you create a tube. By bending the tube and combing the edges labeled “2” you get a torus.

So the square diagram we start with can be thought of as a 3-d object, specifically a torus in this example.



B. Words:

Consider the above torus planar model. We can further simplify the model by generating a ***word*** from the planar model.

The above model could be read as the *word* $2^{-1}1^{-1}21$.

The first 2 and first 1 are notated with a superscript to denote ***orientation***, the arrows run counter clockwise and are therefore inverted.

Any model can be read in a similar fashion. Start at some point and read the labels and their orientation from that point going clockwise.

A ***surface*** is a word which has edges in exact pairs.

And all surfaces are built by combining basic surfaces together.

The basic surfaces are: Sphere, Tours, Projective Plane and Klein bottle.

(Note: Sphere = xx^{-1}, Torus: $ab^{-1}a^{-1}b$, Projective Plane: xx and Klein = $ab^{-1}ab$)

C. Manipulation:

There are 5 rules which we may use to manipulate a word without changing the surface.

(for example: $2^{-1}1^{-1}21$ is the same as $1^{-1}212^{-1}$. The word may be slightly different but it's still a torus. Which brings us to the first, most basic rule.)

1. Cycle rule:

You may change which edge you start reading the word at without affecting the surface.

2. Flip Rule:

You may read a word backwards provided you reverse the orientation of each edge.

3. Cylinder Rule:

The cylinder rule states that a block may be cycled if it is between two letters of opposite orientation.

Let A,B,C,D be blocks (a ***block*** is a collection of edges).

Let x be an edge label.

Example: The word $AxBCx^{-1}D \sim AxCBx^{-1}D$

Where ~ means ***homeomorphic*** or the same.

4. Sphere Rule:

The sphere rule states that you may remove a sphere from a word without affecting its surface.

Let A,B,C,D be blocks.

Let x be an edge label.

A sphere is represented as the word xx^{-1}**.**

Example: $Axx^{-1}BCD$ ~ ACBD.

This actually brings up the idea of ***connected sums***.

A ***connected sum*** is made by taking surfaces and concatenating their words. (The notation is as follows ABC # D = ABCD)

5. Mobius Rule:

The mobius rule is best shown by example first.

Let A,B,C,D be blocks.

Let x be an edge label.

Example: The word AxBCxD ~ $AxxC^{-1}B^{-1}D$

If a pair of edges share the same orientation you may move them next to each other and apply the flip rule on the block of letters on the inside which in turn are put to the right of the mobius pair.

(III)Reduction Algorithm:

Using the above rules we can assemble an algorithm which does exactly what we need it to.Which as a reminder is to take a word and classify it as either a connected sum of Tori or Projective Planes.

Example: Consider to word: ab`$^{1}$cd`^{1}abcd

= a(b`$^{1}$cd`^{1})abcd

[Mobius rule]~ aa(dc`^{1}b)bcd

we may express this as a sum of a plane and some word (aa)dc`^{1}bbcd

~ P # dc`^{1}bbcd

= d(c`^{1}bbc)d

[Mobius rule]~ dd(c`$^{1}$b`^{1}b`^{1}c)

we may express this as a sum of a plane and some word (dd)c`$^{1}$b`^{1}b`^{1}c

~ P # c`$^{1}$b`^{1}b`^{1}c

= (cc`$^{1}$)b`^{1}b`^{1}

[Sphere rule]~ b`$^{1}$b`^{1}

we may express this as a sum of a plane and some word (b`$^{1}$b`^{1})

~ P

Original Form: P # P # S # P

P # P # P

As you can see the algorithm works by finding a basic surface within the word and reducing the word by removing it and repeating the process until all we have are basic components.
Once we have reduced the word completely we may classify it.
The above word (ab`$^{1}$cd`^{1}abcd) is the same as 3 connected projective planes (P # P # P).

Another example below:
= ab`$^{1}$(c`^{1}a`$^{1}$d`^{1})bdc
[Cylinder rule]~ ab`$^{1}$(a`^{1}d`$^{1}$c`^{1})bdc
= b`$^{1}$a`^{1}(d`$^{1}$c`^{1}bdc)a
[Cylinder rule]~ b`$^{1}$a`^{1}(bdcd`$^{1}$c`^{1})a
we may express this as a sum of a Torus and some word: (ab`$^{1}$a`^{1}b)dcd`$^{1}$c`^{1}
~ T # dcd`$^{1}$c`^{1}
we may express this as a sum of a Torus and some word: (dcd`$^{1}$c`^{1})
~ T
T # T