Ch2notes

2020年10月13日 星期二 19:04

Note: (*) means skip in the presentation.

All surface means surface without boundary.

I. Surfaces.

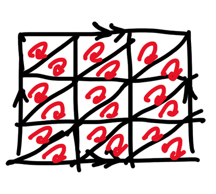
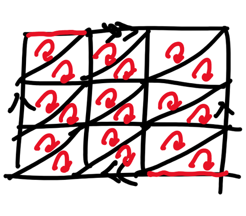
△ Definitions and Facts

Classification of closed surfaces.

Triangulation.

Def: Two triangle share an edge are consistently oriented if they induce opposite orientations on the shared edge.

Thm: M is orientable $\Longleftrightarrow$ The triangles can be oriented in such way that adjacent pair are constently oriented.

Example: T^2:  Klein bottle:  not consistent

Def: For a finite triangulation of S, $n = \#$ vertices, $m = \#$ edges, $\ell = \#$ triangles. The Euler characteristic $\chi(S) = n - m + \ell$

Thus we have $\chi(gT^2) = 2-2g$ $\chi(gP^2) = 2-g$, g is genus.

△ Doubling (*).

The compact non-orientable surface can be obtained from the orientable surface by identifying points in pairs.

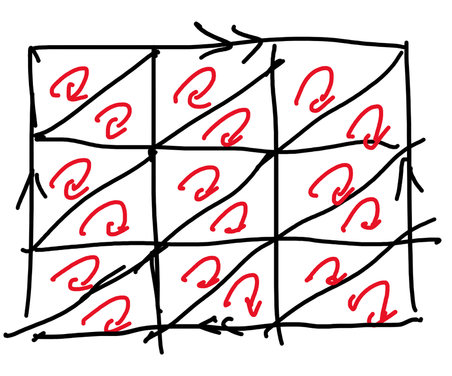
Example: $S^2/\{x \sim -x\} = \mathbb{R}P^2$.

Want to do the other direction;

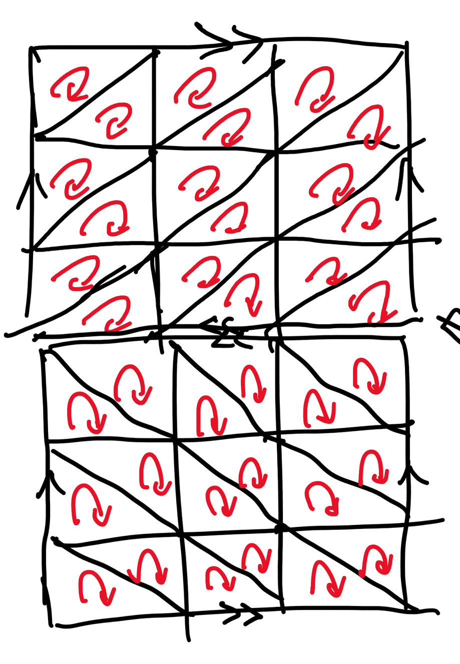
Let N = compact connected non-orientable surface.

Get a triangulation $\longrightarrow$ make two copy of each triangle, edge, vertex. $\longrightarrow$ get connected surface M, M is orientable since one side is consistently facing N.

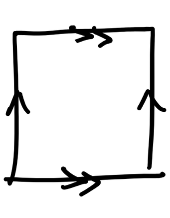
Example: Klein bottle



Double



identify these two edges get consistently oriented triangles.

That is  T^2.

Since all triangles, edges, vertices doubled, $\chi(M) = 2\chi(N)$

$2-2g(M) = 2(2-g(N)) \Rightarrow g(M) = g(N) - 1$.

II. Classification.

Goal: Determine the surface from the given triangulation. K.

Observe: each triangle has 3 edges, each edge belong 2 triangles. So $3\ell = 2m$

$$\Rightarrow \begin{cases} 2n-\ell = 4-4g & \text{if orientable} \\ 2n-\ell = 4-2g & \text{if non-orientable} \end{cases}$$

If already know # vertices n, suffices to get orientability and ℓ to determine the surface.

Genus

```
ℓ = Triangles;
if isOrientable  then return (ℓ-2n+4)/4
                 else return (ℓ-2n+4)/2.
endif;
```

Now, want to write the function Triangles and isOrientable

View triangulation as a graph, triangles = nodes (μ, τ)

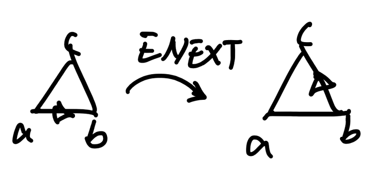
μ is a pointer

τ = 0, 1, 2, 4, 5, 6 order of a triangle

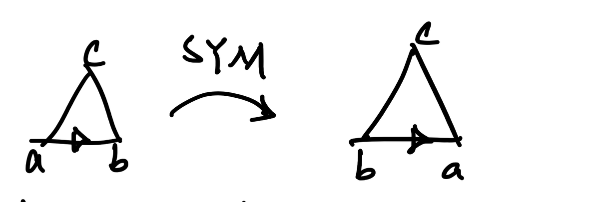
abc bca cab bac cba acb

Here abc means 

```
ENEXT (μ,τ)
    if τ ≤ 2 then  return (μ, (τ+1) mod 3)
             else  return (μ, (τ+1) mod 3 +4)
    end if;
```

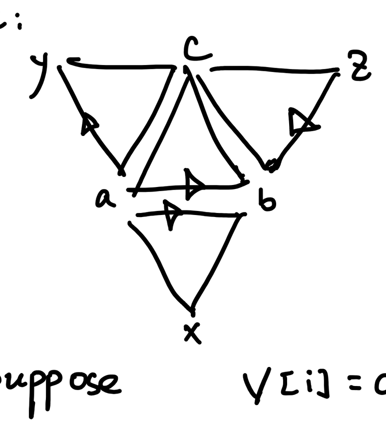


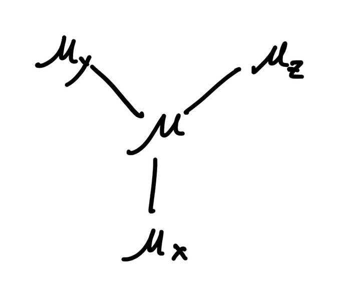
```
SYM (μ,τ)
    return (μ, (τ+4) mod 8)
```



Let $V[1, \dots, n]$ array store vertices of triangulation K

Example:



Graph: 

Suppose $V[i] = a$ $V[j] = b$ $V[k] = c$.

$(\mu, 0).org = i$. This means the first vertex in the label $\tau = 0$ in triangle μ.

$(\mu, 0).fnext = (\mu_x, 0)$ This means a triangle share with an edge μ, the edge given by first two vertex in label $\tau = 0$ in triangle μ.

More examples: $(\mu, 1).org = j$ $(\mu, 1).fnext = (\mu_z, 1)$.

We will use Depth-First Search (DFS) to search the graph.

```
VISIT (μ)
    if μ is unmarked then mark μ; P1;
        for all nbhd ν of μ do
            VISIT (ν)
        end for; P2;
    else P3;
    end if;
```

Here P1, P2, P3 are operations need to do.

For orientability,

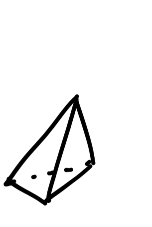
we orient μ at P_1, return boolean at P_2 and P_3, P_2 = whether found contradiction, P_3 = whether have consistent orientation.

```
boolean isOrientable (μ,τ)
    if μ is unmarked then mark μ and choose orientation contain τ.
                                                  Store the orientation.
        bx = isOrientable (SYM(μ,τ).fnext)
        by = isOrientable (ENEXT(SYM(μ,τ).fnext))
        bz = isOrientable (ENEXT²(SYM(μ,τ).fnext))
        return bx and by and bz.
    else return (compare marked and induced)
    endif;
```

Here $SYM(\mu,\tau).fnext$ gives the nbhd of μ.

For triangles,

```
Int Triangle (μ,τ)
    if μ is unmarked then mark μ
        ℓx = Triangle (SYM(μ,τ).fnext)
        ℓy = Triangle (ENEXT(SYM(μ,τ).fnext))
        ℓz = Triangle (ENEXT²(SYM(μ,τ).fnext))
        return ℓx + ℓy + ℓz
    else return 0
    endif;
```

Note: We can view $S^2 \simeq$  and use this as an example to compute orientability and triangles to be familar with the programs.