

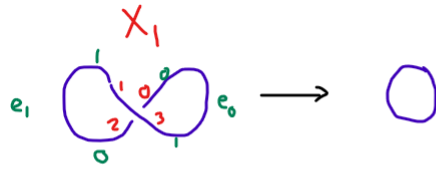
Hello World!

1 Spatial Graph Diagrams

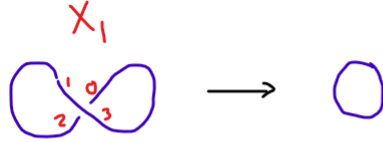
1.1 Reidemeister Moves

1.1.1 Reidemeister 1

Reidemeister 1



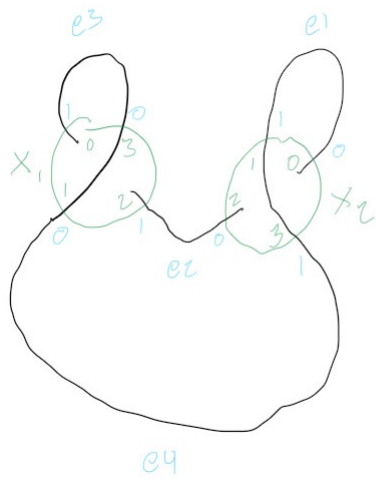
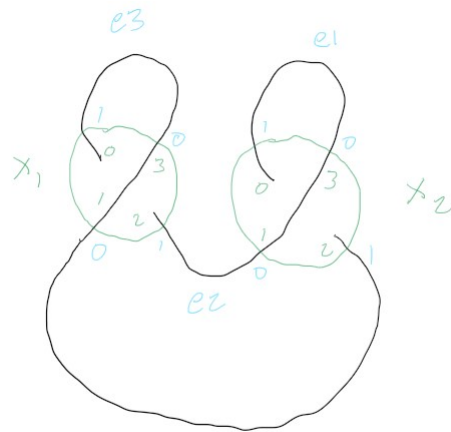
```
1 def test_r1_1():
2     a = pari('A')
3
4     yp_ground_truth = -a ** 2 - a - 1
5
6     x1 = Crossing('x1')
7     x1[1], x1[3] = x1[2], x1[0]
8
9     e0, e1 = Edge(0), Edge(1)
10
11     e0[0], e0[1] = x1[0], x1[3]
12     e1[0], e1[1] = x1[2], x1[1]
13
14     sgd = SpatialGraphDiagram([x1, e0, e1])
15
16     yp_before_r1 = sgd.normalized_yamada_polynomial()
17
18     assert yp_before_r1 == yp_ground_truth
```



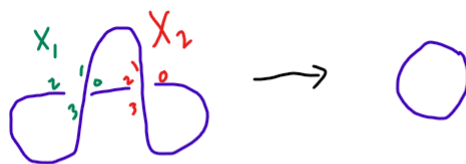
```

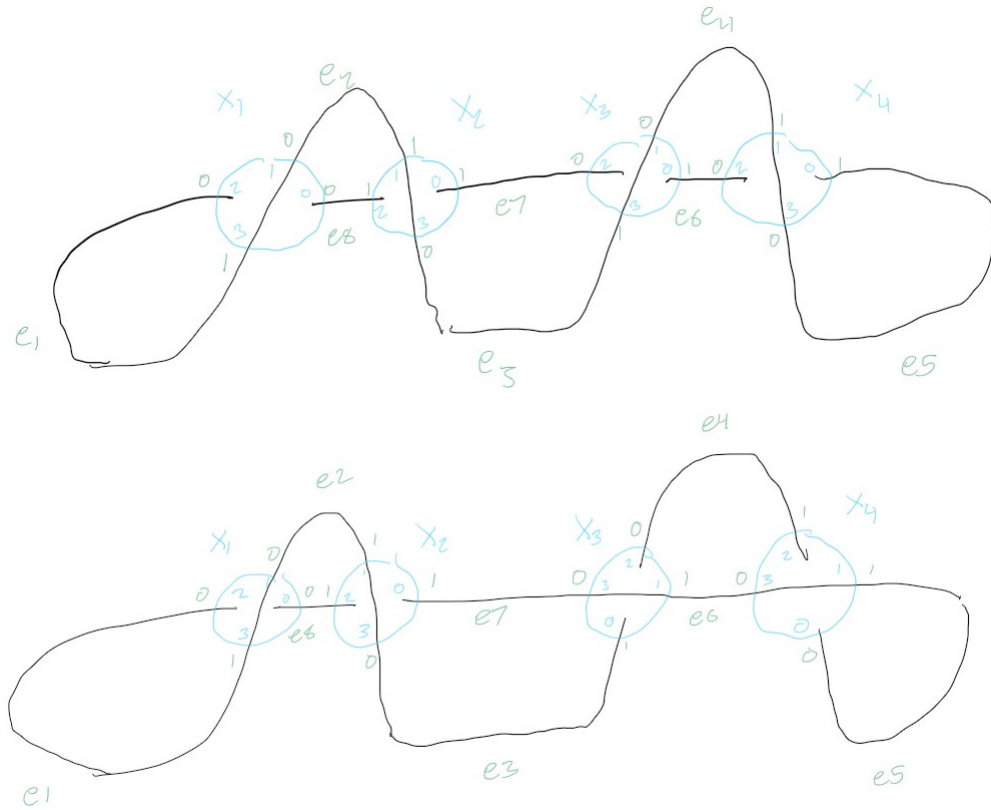
1 def test_r1_2():
2     a = pari('A')
3
4     yp_ground_truth = -a ** 2 - a - 1
5
6     x1 = Crossing('x1')
7     x1[1], x1[3] = x1[2], x1[0]
8
9     sgd = SpatialGraphDiagram([x1])
10
11     yp_before_r1 = sgd.normalized_yamada_polynomial()
12
13     assert yp_before_r1 == yp_ground_truth
14
15     r1_crossing_labels = has_r1(sgd)
16
17     assert len(r1_crossing_labels) == 1
18     assert 'x1' in r1_crossing_labels
19
20     sgd = apply_r1(sgd, 'x1')
21
22     yp_after_r1 = sgd.normalized_yamada_polynomial()
23
24     assert yp_after_r1 == yp_ground_truth

```



1.1.2 Reidemeister 2





1.1.3 Reidemeister 3

The Reidemeister 3 Move

Prerequisites: 1. The SGD has at least one face whose 3 vertices are crossings. 2. The candidate face has at least one edge that pass completely under or over the other two edges.

Algorithm: 1. Identify the candidate face for the R3 move. - If there is more than one face that satisfies the R3 prerequisites, then choose one (TODO Implement choosing logic).

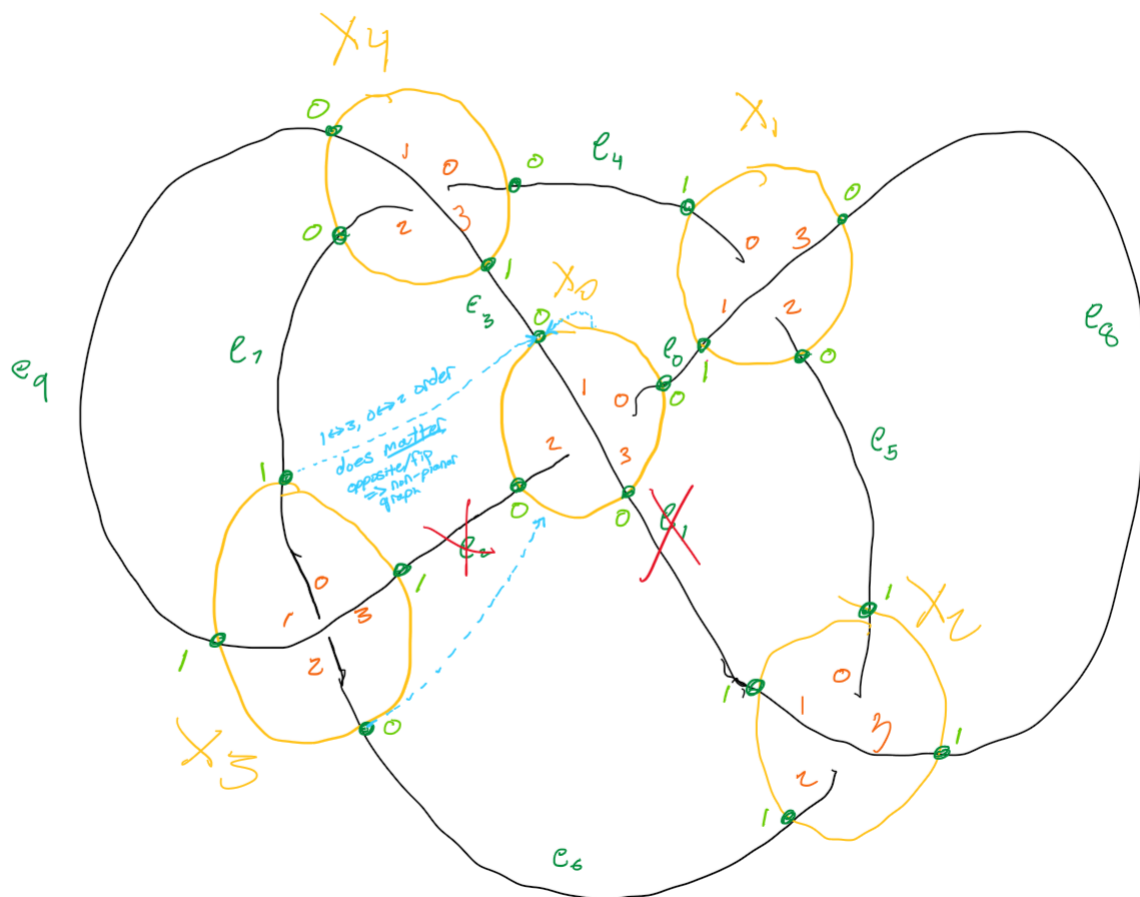
2. Identify the candidate edge for the R3 move. - If the chosen face has more than one edge that passes completely under or over the other two edges, choose one (TODO Implement choosing logic).

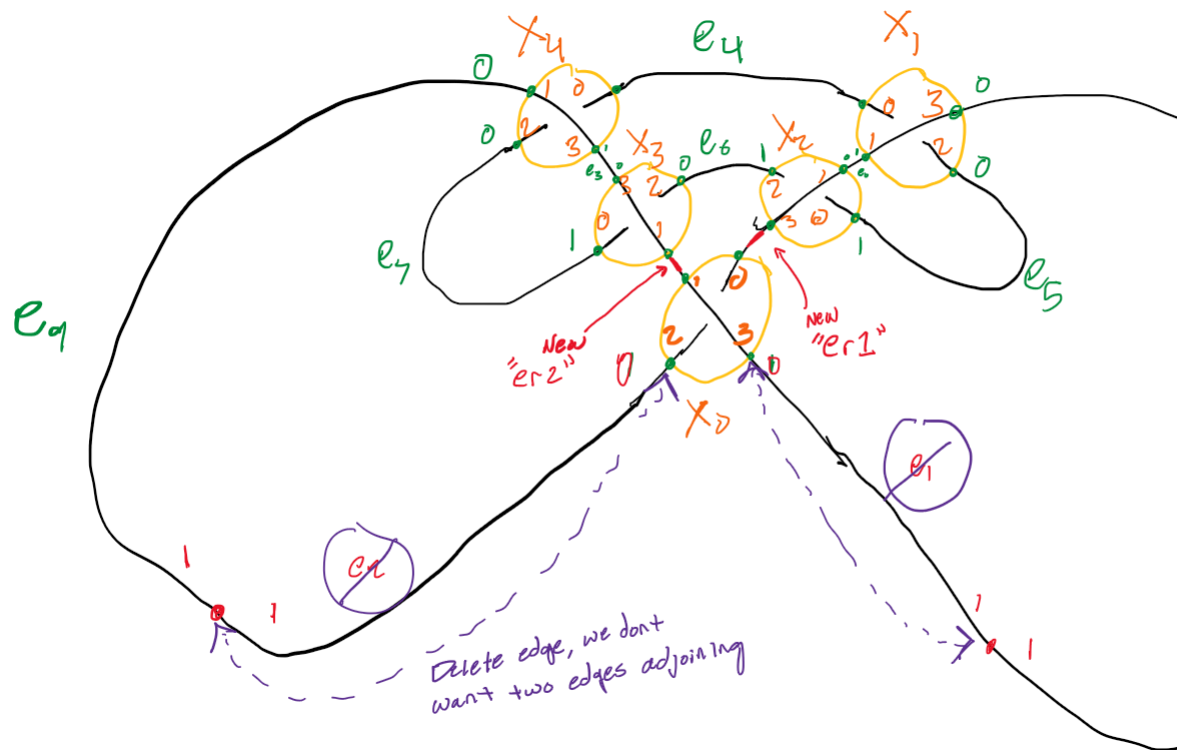
3. Label each vertex and edge of the candidate face. - The R3 moves one edge across its opposing vertex. - The chosen edge will be called the "moving" or "opposite" edge since it will move across the opposing vertex. - The vertex opposing the moving edge will be called the "keep" crossing

since it will remain in place. - The two edges adjacent to the keep crossing will be called "common" edges (as opposed to the opposing edge). - The two other crossings will be called "remove" crossings since moving the opposite edge will remove these crossings (and introduce two new crossings on the other side of the keep crossing).

4. Identify how edge and crossing indices will change by the R3 move.
 - The moving edge intersects with two keep crossing edges. The R3 move will cause it to intersect the opposite two edges (i.e.). - When we remove the two remove crossings, two edges represent the moving edge and will be moved by the R3 move. The other two edges must spliced together. - ...

5. Perform the R3 move. - Delete the two edges that do not move. - Connect ... - Shift the two move crossings - Add new edges... -





1.1.4 Reidemeister Simplification