Performing Verification of elle and ~elle equivalence classes

Vadim Korotkikh

[va.korotki@gmail.com](mailto:va.korotki@gmail.com)

Calculating Vij Holoraumy Matrices from Vierergruppe left cosets

Performing the verification is a multi step process. The first way that the verification was performed was by taking all the left coset groups of Vierergruppe via S3 and hardcoding them in Python Numpy.



Hardcoding and doing the calculations. vprime is plain *V*



Each Vierergruppe flop is according to first example under C Permutations in appendix and its accomplished by the above code.

Vierergruppe flop: (123)V  
matrix([[0, 0, 1, 0] matrix([0, 0, 0, 1] matrix([1, 0, 0, 0] matrix([0, 1, 0, 0]

[1, 0, 0, 0] [0, 1, 0, 0] [0, 0, 1, 0] [0, 0, 0, 1]

[0, 1, 0, 0] [1, 0, 0, 0] [0, 0, 0, 1] [0, 0, 1, 0]

[0, 0, 0, 1] [0, 0, 1, 0] [0, 1, 0, 0] [1, 0, 0, 0]

Vierergruppe flip are also hardcoded.



From this, each Vierergruppe is calculated, first the flops and then for each flop all the flips. The flips are calculated the same way as in Appendix D, pg 23 in the paper.



The resulting flopped and flipped Adinkras are sent to ~Vij holoraumy matrix calculating function.



Where ~V is



To simplify calculation, I multiplied the above equation by 2i, removing the i component and flipping all the signs. This allows for quicker cross-referencing of calculated ~Vij matrices because in Python dealing with i in matrices is messy. For example, alpha^1, alpha^2 and alpha^3 become the following

2i \* alpha^1

[[ 0 0 0 2]

[ 0 0 2 0]

[ 0 -2 0 0]

[-2 0 0 0]]

2i \* alpha 2

[[ 0 2 0 0]

[-2 0 0 0]

[ 0 0 0 2]

[ 0 0 -2 0]]

2i \* alpha 3

[[ 0 0 2 0]

[ 0 0 0 -2]

[-2 0 0 0]

[ 0 2 0 0]]