New Gadget Verification Code Description

Vadim Korotkikh [va.korotki@gmail.com](mailto:va.korotki@gmail.com) November 2017

This document write up contains the information regarding the mathematical algorithm to calculate the New Gadget values over the entire space of BC4 36,864 Adinkras and how it is written and executed using Python 3. Specific Python version used in calculation was Python 3.5, but the code is also compatible with Python 2.7.

Software wise the code builds upon earlier developments/works by the author but with changes to the code that pertains to the final gadget calculation. To speed up Gadget calculation, multiprocessing feature has been added and is utilized within the code. The code also now produces a text output of the results which can be zip compressed for distribution/sharing of results.

Using the elemets of BC4 coxeter group, the script adinkra\_nxn\_constructor.py builds the the 384 L sign permutation matrices (4x4, row x col). These L sign permutation matrices serve as the building blocks of all Adinkras, given that any two of them satisfy conditions of a set Garden algebra equations . The same script also handles the process of assembling all possible Adinkras using the L sign permutation matrices. Using 384 L matrices, the script creates 36,864 Adinkras with N=4 (4 colors), four open-nodes and four closed nodes

For calculating the Fermionic Holoraumy matrices the script fx\_vij\_holoraumy.py is used. The actual function to do so is calc\_fermi\_vij( adinkra list) and it takes in a ‘list’ or an array of pointers to arrays containing 4 L matrices defined using numpy np.array. Using the 4 L sign permutation matrices, the V~tilde Holoraumy matrices, V12, V13, V14, V23, V24, V34 are calculated for each Adinkra. Each set of six V~tilde Holoraumy matrices are appended to a list which then gets appended/added to the final return list. This return list, called vij\_fermi is passed back to it’s execution origin in cls\_adinkra.py script which serves to declare and describe AdinkraSet class. AdinkraSet class is used to executing various Holoraumy functions and storing the return results to be used by other functions.

Because there is two ways of calculating Gadget values this includes calculating the new Gadget there is also a function called calc\_vij\_alphabeta( main\_tetrad\_list) which is also found in fx\_vij\_holoraumy.py script. This function takes in the adinkra list, calculates the V~tilde Holoraumy matrices and then compares them to < alpha or beta > matrices ( which are actually outer product combinations of Pauli matrices. Each V~tilde Holo is checked to see whether it’s a coefficient factor of one of the three Alpha or three Beta matrices, which if true provides with a positive or negative factor of 1, ie the tilde ell coefficient integer. Because there are six V~tilde Holo matrices, there are six coefficient integers which are added to a list of six for each Adinkra and in the end added to a final return list.

Both the V~tilde Holoraumy matrices and the coefficients can be used to calculate the New Gadget values. In terms of computation complexity and speed, given the fact that there are 36,864 Adinkras to go through, using the alpha/beta coefficients proves to be a much faster way.

The calculations were done using both methods, using the beforementioned functions as generators. The fx\_mpgadgets.py script contains the codes for performing the New Gadget calculation. The mp\_gadgetcalc\_abonly function performs the New Gadget calculation using only the alpha-beta coefficients values list. The function employs multithreading module in order to maximize computational speed. The function also contains the output writing code, that writes the calculated Gadget values to a text file for each Adinkra.