Journal

Son Nguyen

October 05, 2010

The goal for the quarter is to write code to fill out some details in the paper and automatically generate code that would actually compute the formulas in the paper.

1 First to second Week

During the first 3 weeks, I tried to better understand the goals of our group. I have written the following code:

- Code to generate all the possible case of geminals, without removing duplicates. The rule for this generation is generate all the combination of 3 pair (a,b),(m,n),(u,v) where a, b, m, n, u, v can vary from 0 to 5. There are 27,000 cases. If we restrict a¡b, m¡n, and u¡v the number of cases reduce to 3375. Since (a, b, m, n, u, v) present the variables, and in some cases, there are less than 6 variables in the expression, we create the skipping rule to reduce the case $((0, 1), (0, 2), (4, 5))$, which is missing 3, to $((0, 1), (0, 2), (3, 4))$, and so on. By applying this rule, the number of cases narrow down from 27,000 to 3272, and from 3375 to 409.

2 Third Week

In the third week, we start a new approach. We again generate a combination $((a, b), (m, n)(u, v))$; however this time each pair in the combination represent the functions w_1, w_2, and w_3 respectively. First, we generate all possible combinations. The rules for this initial generating is:

- $(a, b) = (0, 1)$
- $m \neq n$, and $u \neq v$,
- $m = crea(a, b)$; $n = crea(a, b, n)$; $u = crea(a, b, m, n)$, and $v = crea(a, b, m, n, u)$. Where the function $crea(x)$ returns a list of all elements in x plus length of $x + 1$.

Using these rules, I wrote a code, and get 87 cases in total.

3 Fourth Week

- This week, I continue with the code in the previous week. The 87 cases need to be reduced by removing duplicates. The first rule to apply is ordering each pairs, for example: $(0,1)(1,0)(2,0) \rightarrow (0,1)(0,1)(0,2)$. After applying this rule, all same combination will be grouped and numbering the number of combination in each group. I have written a code to appply this 2 rules and get 29 groups from 87 combination.

- The second rule is to interchange 0, and 1, then apply the ordering in the first step. However, I did not find any duplicates after applying this rule. If we order the combinations, for example: $(0,1)(1,2)(0,2) \rightarrow (0,1)(0,2)(1,2)$ then I can see some duplicates.

4 Fifth Week

- I continue the code left in the fourth week. I figured the bug when applying the shifting $0 \leftrightarrow 1$, and fixed it. Now, after the shifting, and ordering the number of cases has been down to 18.

- At first, I have difuculties presenting the idea of the algorithm into the code. The algorithm in this case is quite clear: We pick up the first element and checking it with the rest of the list after transformed. After the checking process, we delete that first element and all the elements that found to be duplicate. And keep doing this until there is no elements left in the list. However, when it comes to the code, I have trouble choosing the suitable varibles, statements and how to organize them on the loop. I have learned couple of things on handle those kind of situations.

5 Sixth Week

- Continuing the code in the firth week. The cases 18 still need to be reduced. The new rules is exchange 2 to 3 and find the duplicates. Here is the procedure in my code:

  - Find all the elemements (in 18 elements)which contains both 2, and 3
  - Switch 2 to 3 in those elements
  - Order each pair in those elements
  - Use the simliar algorithm in fifth week to remove duplicates

- After this procedure, I ended up with 16 cases, which is the same as the result Dr Martin has.

- The 16 cases is for the geminal with 6 varible: $[[a, b], [m, n], [u, v]]$. For the geminal of 5 variables: $[[a, b], [m, n], [v]]$, we do exactly the same algorithm to find out the number of cases. To make the most use of the code for the 6 variable geminal, I put the imaginary number 2010 to the geminals of 5 variable, making it become $[[a, b], [m, n], [u, 2010]]$. The result I obtained is 9 cases in total, which also the same as Dr. Martin's result.

6 Seventh Week

- We start the algorithm to put the loop for each geminals have created. Here is the tasks that I need to do:
  - Rewrite the code to symbolically compute the matrix's determinant including the sign for each elements
  - Write the function to get the group symmetry. The group symmetry elements indicate the shifing that make the geminals (without loop) unchanged.
  - Use these 2 codes, combining with the regular "'removing duplicate algorithm"' to get the number of cases for all geminals
- So, I wrote the code to get the new symbolic determinant. I have adjusted the former determinant, by adding one a entry containing the sign in each elements of the former determinant. The code was tested by some small matrix and did not see any problems. I believe the code creating new symbolic determinant is correct.

7 Eighth Week

- Continuing with the second step of the plan I made in seventh week, I wrote the code to create the group symmetry for all 16 geminals of the case $[[a, b], [m, n], [u, v]]$. The group symmemtry I obtained did not have many elements. There are 8 of them has only one identity element. The largest group symmmetry is of the geminal $[[[0, 1], [2, 3], [4, 5]], 1]$ with 8 elements.
- Also, I spent a lof of time to create the group symmetry for each elements of the determinant, which was, unfortunately, not nessesary. However, the result does not be affected since the 2 group symmetry are identical.
- I did the last step in the plan, which was write the code to remove duplicates in the expansion of the determinant for each geminals. However, after running the code for all 16 geminals, there was only 1 duplicate was found! That was understandable since there only few elements in most of the group symmetries.

- After discussing with Dr Martin, I figured out that the outcome was not correct, and there was some bugs in my program. I checked and discovered that I did forget to order the determinant after expanding it. By redoing this part, I came up with a new result, in which many duplicates was found.

8 Nineth Week

- I wrote a report for a final presetation on Wednesday. At first, I attemped to write it on Texshop on MacOSX. However, I had a lot of problems with its command, interface, and sometime I could not run a Latex code. After a while, I switched to TechnichCenter, and finished the first draft for a report.

- Also, I put all the descriptions to the code that I had done so far. There was still some points not so clear, but I would work on it from now on.

9 Tenth Week

- I wrote a similar code to for geminals in the form $[[a, b], [m, n], [u]]$. At first, I encountered some difficulties with the "'imaginary number"' 2010 that I created before for those geminals. By tracking down all the function, I figured out where I need to change so that the old code can be used again. I endded up producing the similar result for all (9) geminals in this form.

- I continued putting some more descriptions to the code that I wrote. There was still some part in the code need more clear descriptions.