

Journal
Santosh Vummidisingh
FALL 2010

Our Initial goal for the quarter was to understand the given paper completely and then eventually write a code that will automatically generate the code.

First 2 weeks

During this time I mainly tried to understand what exactly our research paper was all about. It was a little difficult because the terminology used in the paper was not something I had seen earlier. Comprehending what to do was a little difficult during this stage. Our group was assigned the task of generating all the different possible cases that can be generated with the geminals during this stage. We ended up generating 27000 different cases with the condition that the 3 pairs $(a,b), (m,n), (u,v)$ where $a; b; m; n; u; v$ can take the values from 0 to 5. If the restriction of $a \neq b, m \neq n,$ and $u \neq v$ was taken into consideration then the number of cases reduced to 3375. Sometimes there might be less than 6 variables and in such cases we had 409 cases.

Third Week

We considered the functions $w_1; w_2;$ and w_3 and represented them in the form of $((a; b); (m; n)(u; v));$ Using the constraints, $(a; b) = (0; 1) m \neq n;$ and $u \neq v;$ We generated the different combinations using $\text{crea}(x)$ which is of length $x+a;$ and contains all the different elements. By the end of this week we had 87 cases. At this stage, I made a presentation in our weekly group meeting regarding our progress.

Fourth Week From our code group meeting this week, we realized that there are quite a few duplicates in these 87 cases and we tried to figure out how to eliminate them. In order to achieve this task we had to order all the pairs in the ascending order. Similar combinations were now grouped together and 29 such groups were found.

Fifth Week

We now interchange 0 and 1 with each other and then applied this ordering to the earlier step. This was not particularly fruitful and after we ordered the combination (Eg: $(0; 1)(1; 0)(2; 1) \neq (0; 1)(0; 1)(1; 2)$) and in this way some more duplicates were removed. After applying all these different ordering techniques we now arrived at 18 cases.

Sixth Week

From the hand written cases that Dr. Martin already found out, it was clear that there were still a couple of duplicates in the 18 cases as well. Son then came up with a logic wherein he found all the elements that contained the elements 2 and 3. He then swapped 2 and 3 with each other and ordered them. By doing this we arrived at 16 cases.

Weeks 7 and 8

Over the past 2 weeks we have found all the duplicates that are possible. These cases did match with ones that Dr. Martin had already discovered. During the last one week we have looked at the different ways in which we can store and generate determinant terms of the matrix under consideration for each of the final 16 cases that we have arrived at by making use of the reduction properties.

I have also attended the responsible conduct of research (RCR) workshop. This workshop has helped me learn a lot of new things about the ethical standards, responsibilities, conflicts of interest, etc with regard to research. This seminar has definitely widened my perspective about research work.

Dr. Martin asked me to generate the code for the group symmetry and I was unsuccessful in generating what was required. Son eventually generated the code for the group symmetry.

I personally started giving a lot more time to understand the programming concepts in Python. I have been viewing and reading a lot of online tutorials to help me grasp the coding methods. I am able to write only simple functions as of now. Learning the syntax in Python has probably been the greatest challenge for me but I am positive that I will be able to make greater contributions in the weeks ahead.

Weeks 9 and 10

Our final report was made and a presentation was done during week nine. Also, Dr. Martin assigned me the task of running a function that will take sons output as an input to my function and return a couple of outputs. The different outputs that were required were to check the if there was a loop on any of the elements, if there was loop then our function had to not only identify it but it also had to remove the germinal which had that loop. The initial code that I wrote did removed the element but it did not mention which element was removed and Dr. Martin advised me to implement this requirement in the final output . I then wrote the code that will implement this task but it generated a couple of errors. I was getting confused about where exactly I had to call the required function. More often than not there was a parameter mismatch and the output was not being generated.