**Objective and summary of work done**:

Understanding Large scale software systems with typical complex interacting elements like packages/classes/interfaces/inheritances/methods/attributes is a time taking process and cause maintenance of the code challenging task. The main objective is to provide the technology which would find the structure and importance of the classes and packages in large complex source code base so that maintenance personal time to understand the code would be reduced and help further to change the code structure better to improve the performance.

The Element ranking is enhanced formula of Page ranking to get the rank for each element and then aggregate with the Analytical hierarchy process. The weighted page rank value of each element class/package of any layer will be aggregated. This process involves the steps 1) read the structure of the code and prepare the network layer graphs for each relation 2) calculate the page rank of element and aggregating the weighted page rank with analytical hierarchy process

Relations found are:

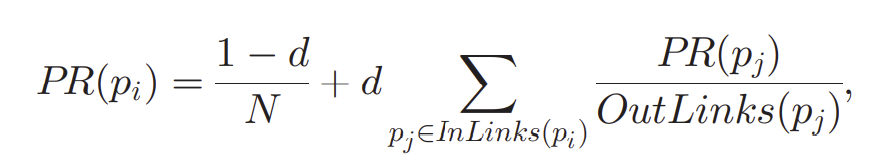
1. Inheritance INR
2. Implements IMR
3. Parameter PAR
4. Global variable GVR
5. Method call MCR
6. Local variable LVR
7. Return type RTR

The relative importance between each relation is taken from the paper W = (0.034, 0.290, 0.034, 0.034, 0.178, 0.394, 0.034)

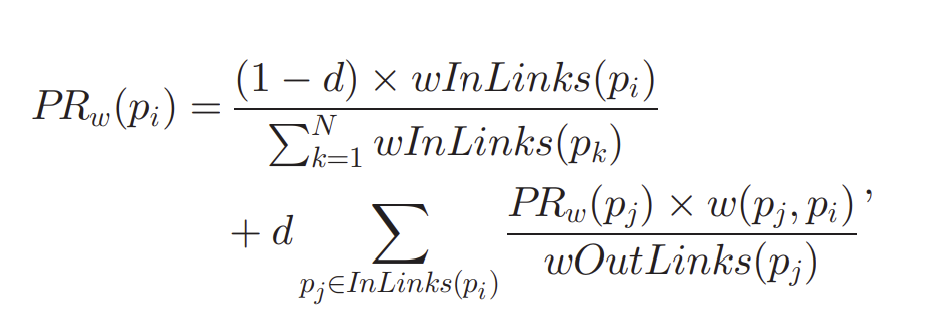
**Comparison with existing work**:

The actual page rank algorithm enhanced to consider the weighed graph as below

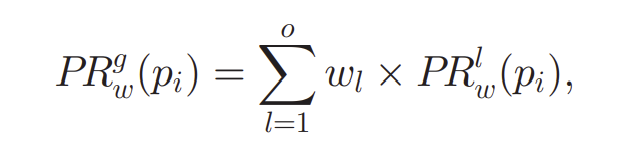
Page ranking:



Weighted page Rank:



Aggregating weighted page rank Or Global weighted page rank:



As part of algorithm testing, we have taken the java jdk code MPN network to compute the global weighted page ranks.



The original ranks provided for the same data set from paper

