

Readme

Implementation:

I have Implement 2 Algorithm from Call Graph (CHA and RA).

I Implement CHA(**Class hierarchy analysis**) The pseudo code for this algorithm is like bellow:

for each method

 for each callsite

 if name_of_callsite =name_of_callsite in other method which is in hierarchy with method

 create an edge between new method and method

I use Atlas to have benefit of visualization of Graph which is very helpful in debugging.

The return of the program is a text file with this format

number of nodes, number of edges, number of virtual callsite, max, min, average

number of calls linked to each virtual callsite

HowToRun

map java test case in Atlas.

create an Atlas project and put these two file in a package with name of CallGraph. then

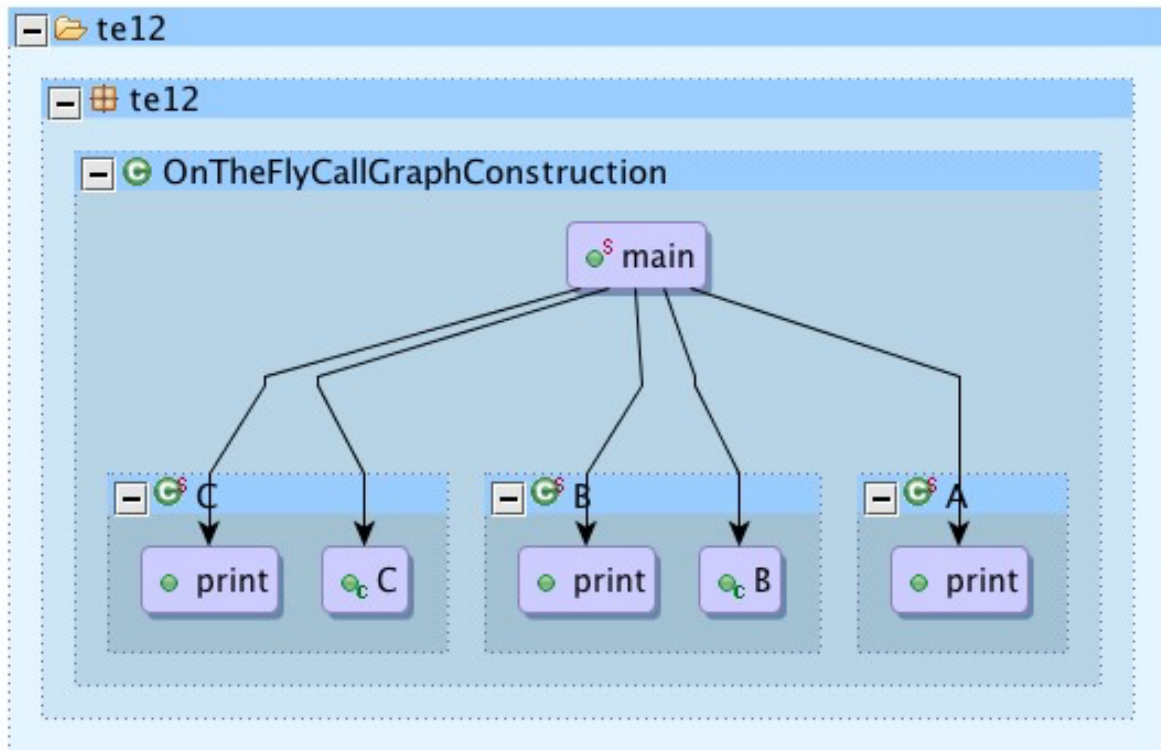
use Atlas shell :

CallGraph.CHA.Run(false)

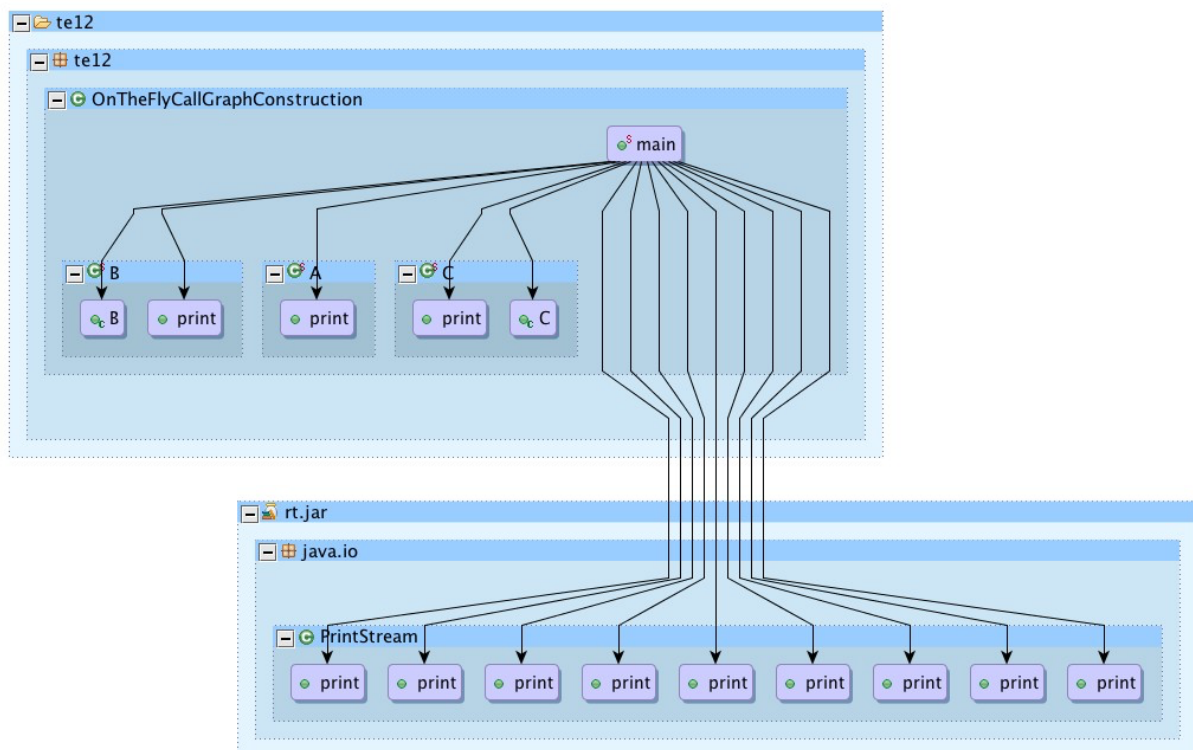
var C = Common.universe().edgesTaggedWithAll("CHA").retainEdges();

show(C)

CallGraph.CHA.WriteInFile()



I implement the RA algorithm, The result is like bellow. RA algorithm has more edges.



Not Implemented:

I don't check the signature of method and ignore the type of variable.

Finally my result show smaller number of edge and node in CHA in compare of RA which is completely reasonable.