



GS04: Tools and Environments

Coursework

To be handed in at MPEB 5th floor reception by noon Feb 18th, 2008

The aim of this coursework is to develop an Eclipse plug-in that will analyze a project and calculate the cyclomatic complexity of every method defined in any class of the project.

The cyclomatic complexity of a program block is defined in "A Complexity Measure" by T.J. McCabe in IEEE Transactions Vol SE-2(4):308-320. December 1976.

In this paper McCabe defines cyclomatic complexity as

$V(G) = e - n + p$ with

e – number of edges in the control-flow graph G

n – number of nodes in the control-flow graph G

p – number of connected components in the control-flow graph G

Thus to calculate the cyclomatic complexity of a method you will have to build a static analyzer that computes the control-flow graph of the method's body. Nodes in this control flow graph represent individual statements. A node $n1$ is connected with a node $n2$ by an edge if it is possible for the statement represented by $n2$ to be executed immediately following the statement that is represented by $n1$. In structured programs that are free of goto statements there is always only one connected component in the control flow graph. Thus you may assume p to be 1 when analyzing Java programs.

To test your plug-in use the test-data provided at

`svn+ssh://collins.cs.ucl.ac.uk/cs/student/misc0/stud/mscssep/2007/repo/w.emmerich/Coursework/trunk`

To hand in your coursework provide

- The URL of your Subversion repository into which you have checked in
 - your commented source code,
 - API documentation
 - unit tests
- The cyclometric complexity measurements for the test data provided

There is probably a large number of Eclipse plug-ins available that perform cyclomatic complexity analysis. I am only interested in your one and would therefore like to remind you **of the departmental plagiarism policy available at http://www.cs.ucl.ac.uk/teaching/msccs/msccs_cwrk.html**