

# COMP 621 Program Analysis and Transformations Assignment #2

Using the McLab Framework for Analysis and Profiling

Due: Thursday March 1, 2012

## Overview:

The purpose of this assignment is to give you some practice designing flow analyses, and to familiarize you with the McLab analysis framework, so you will be comfortable with it for your course project. You have been given an introduction to the system during the lecture. Chapter 5 of Jesse Doherty's master's thesis (<http://www.sable.mcgill.ca/mclab/mcsaf/mcsafthesis.pdf>) contains a detailed description of the analysis framework. You can also access an online documentation of the framework's API, with an example, from [http://www.sable.mcgill.ca/mclab/c621\\_special.html](http://www.sable.mcgill.ca/mclab/c621_special.html).

Please feel free to ask questions of the graduate students in my research lab, the TAs during their office hours, and also feel free to share information with your classmates on the google group. The objective of this assignment is to get everyone comfortable with the environment - so don't spend a lot of time getting stuck on some small technical issue.

We will provide everyone in the class with some benchmarks which are known to run correctly with both McLab and AspectMatlab.

## Question 1 - *Implementing an Analysis in McLab*

McLab provides a structured flow analysis framework that can be used to implement both forward and backward analyses. For this question you can choose **one** of the following analyses to implement. However, you are strongly encouraged to suggest your own analysis, just give a clear definition of it. The problems below are listed in increasing levels of difficulty.

- **Live Variables:** Implement an intra-procedural data flow analysis for live variables.
- **IsScalar Analysis:** In MATLAB all variables are, by default, 2-dimensional arrays. However, many variables, for index variables like *i*, *j* or *k* are only ever used as scalar variables. Write an analysis, that determines at each program point, if a variable is: (1) scalar (this variable must be a scalar variable at this point), (2) non-scalar (this variable is definitely not a scalar at this program point), or (3) top (the variable could be a scalar or a non-scalar).
- **IsInteger Analysis:** The base type of variables in MATLAB is double, however often variables will only contain integer values (and hence can be stored more efficiently). Write an analysis, that determines at each program point, if a variable is: (1) integer

(all values stored in this variable are integer at this program point) or (2) top (the types of the values stored in this variable are unknown).

You should implement the analysis and use the pretty printing mechanism to print out the results of your program for several small illustrative examples. Choose the examples to illustrate the subtle points of your analysis and to show that it is correct.

For the purposes of this question you should submit a short description of your approach to implementing the analysis, the source code for your implementation (only those files you implemented), and several small example programs with the pretty-printed analysis results.

## Question 2 - *Profiling*

It is sometimes useful to insert instructions to profile the behaviour of programs. For example, it might be interesting to know how many function calls or array references a program makes. We have already seen that one can use Aspects to implement such profiling. Another way to accomplish this is to directly instrument the IR.

For this question, you will try profiling using McLab and also using AspectMatlab.

(a) **Using McLab:**

Implement profiling of some feature of your choice (for example, function calls, array accesses, loop iterations, flops, etc.).

You should submit a short description of your approach to profiling, your source code, and some examples showing the pretty-printed IR before and after inserting the profiling statements. You should also try your profiler on your benchmark program (from assignment #1), and report the results of your profiler.

(b) **Using AspectMatlab:**

Implement profiling of some feature of your choice using Aspects. This does not need to be the same feature you profiled in (a), but it would be interesting to do the same feature in order to compare the results and the ease of implementing the profiling.

Write a brief comparison of profiling using AspectMatlab compared to an IR rewriting as implemented in McLab. In particular, what would be needed to make AspectMatlab as flexible as bytecode rewriting? On the other hand, what are some advantages of AspectMatlab over a bytecode rewriting tool?

Apply your AspectMatlab profiler to your benchmark program from Assignment #1 and report the results.

If you implemented the same profilings for both parts (a) and (b), report the execution time (using on the production MATLAB system) for each profiler and discuss which is more efficient and if one is more efficient, discuss why.

You should submit a short description of your approach to profiling, your source code, your comparison and evaluation of using AspectMatlab vs. McLab, and the results of applying your profiler.