Course Project: Submit all the files by Email to msagiv@acm.org. Due August ~~2~~nd

A: Solve **one** of the following projects. The TVP files and all the documentation is available from http://www.cs.tau.ac.il/~tvla

Project 1: Proving Partial Correctness of a Simple Mark and Sweep Garbage Collection (gc directory)

1. Document the predicates and the actions in the tvp files for the Mark and Sweep example. In particular, explain the instrumentation predicates and their meanings (files: predicates.tvp, actions.tvp).
2. Remove the focus operations form the following actions: `(AssignFieldSelect, IsNotNull, and IsNull file:` actions.tvp and SetSelectAndRemove from action_set.tvp)`.` Study the resulting analysis. Does it produce results with the same precision? If not then what is the reason?
3. Run the original analysis on an example marking procedure that only traverses the left pointer field. How does the analysis behave?
4. What are the difficulties in extending the analysis to handle Garbage Collection Algorithms like Copy Garbage Collection in which the garbage collector can mutate the heap
5. Add actions for showing that the Mark phase must eventually terminate. One way to show that is by showing that the set of nodes reachable from the pointer variable x (used in the while loop condition in mark.tvp) decreases in every loop iteration.

Project 2: Proving Partial Correctness of Sorting Algorithm (sll_sorting directory)

1. Document the predicates and the actions in the tvp files for the sorting example. In particular, explain the instrumentation predicates and their meanings (files: predicats.tvp, and actions.tvp).
2. Remove the focus operations from the following actions: : `Greater_Data_L`, `Less_Equal_Data_L`, and `Greater_Equal_Data_L (file:` actions.tvp). Study the resulting analysis. Does it produce the same results?
3. Write an improved version of bubble sort in C called smart-bubble-sort which remembers the head and tail of the list which are already sorted and does not traverse them if not needed in the next iterations. Then, convert it manually into tvp and run it. Study the results of the analysis
4. Add actions for showing that the loops in insertion sort and bubble sort must eventually terminate. One way to show that is by showing that the set of nodes reachable from the temporary variable x decreases at every loop iteration
5. Bonus: Give an example of correct sorting procedure that TVLA fails to verify with the existing predicates and actions.

Theoretical questions: (Solve 4 out of 6)

1. Give an example program in the points-to analysis which demonstrates the effect of non-distributive transfer functions. Explain your answer.
2. Show the impact of the non-montonicity of widening in the interval programs where the programmer can specify a precondition at the procedure entry.
3. Give an example program in which the TVLA analysis will require exponential time in terms of the input.

   The Octagon abstract domain is defined in http://www.di.ens.fr/~mine/oct
   You can find documents and a prototype which has web interface.
4. Give an interesting example program in which the Octagon can prove the absence of array bound violations and the algorithm from assignment 2 will not.
5. Show an example program where the Octagon widening leads to loss of precision.
6. Show an example program for which the Octagon analysis is not as precise as Polyhedra. You can find more information on Polyhedra at http://www.cs.unipr.it/ppl/abstractions