CS615 Homework1

Max marks: 50 **Suggested Time: 1 hour**

- *Be brief, complete and stick to what has been asked.*
- *Unless asked for explicitly, you may cite results/proofs covered in class without reproducing them.*
- ***If you need to make any assumptions, state them clearly.***
- ***This homework is a replica of an earlier mid-sem, where the time allowed was more than the suggested time of 1 hour. Solving these problems on your own will give you a good sense of how much you have understood the subject.***
- ***Do not copy solutions from others.***
  Plagiarism is an evil that we must uproot collectively.

1. Consider the following program in a C-like language, in which all variables are of type `int`.

```
L0:  a := 0; b:= 0; i := 0;
L1:  while (a < 5) do
L2:     a := b + 1;
L3:     b := a + 1;
L4:     i := i + 1;
L5:  // end-of-while loop
```

We wish to analyze this program using the interval (or box) abstract domain. Thus each abstract state is a triple (I_a, I_b, I_i), where I_x denotes the interval of variable `x`. As discussed in class, each I_x is either $(-\infty, x_u]$, $[x_l, \infty)$, $[x_l, x_u]$ or $(-\infty, \infty)$, where x_l and x_u are integers with $x_l \leq x_u$.

(a) *[10 marks]* You are required to find the abstract transfer function of one iteration of the `while` loop. In other words, if (I_a, I_b, I_c) is the abstract state at `L1`, and if the loop is entered and executed, what would be abstract state after executing each of the statements at `L2`, `L3`, `L4` once? Your transfer function should give the lower and upper bounds of each variable after executing the statement at `L4` in terms of the lower and upper bounds of variables just before the loop body was entered at `L1`.

(b) *[10 marks]* We now wish to compute the abstract loop invariant at `L1`. For purposes of this subquestion, use the widening operator (without thresholds) for the interval domain as defined in class, but defer widening for two iterations (i.e. do not widen the first two times you are trying to combine abstract intervals at `L1`). Compute as good an abstract loop invariant as you can (narrowing, if necessary), and clearly show all your steps.

(c) The upper bound for the abstract interval for `i` in the loop invariant computed above is an upper bound of the number of iterations of the `while` loop.

i. *[5 marks]* How many times does the `while` loop in the above program actually iterate? Your answer need not depend on the abstract loop invariant, but you need to indicate your reasoning.

ii. *[5 marks]* Does your estimate match the upper bound for `i` in the abstract loop invariant computed by you? If yes, indicate if you would get a similar match if the loop condition was changed to `(a < 5000)`. If no, indicate if you could obtain the desired match by deferring widening for a few more (finite number) iterations. In each case, your answer should be accompanied by brief justification.

2. Student A wishes to come up with a new abstract domain with the intent of analyzing the program in the previous question, and obtaining an exact count of the number of times the while loop iterates. Fellow-student B indicates that the octagon domain would serve this purpose. In the context of the program under consderation, an element of the octagon domain is a 6×6 matrix of integers or ∞, as discussed in class. Professor P comes along and indicates that although B's suggestion is fine, the octagon domain would keep track of more inequalities than are required to obtain the exact iteration count of the while loop.

   (a) *[5 marks]* How many distinct inequalities would A be keeping track of in each abstract state if she were to follow B's advice? Note that this can be less than 36.

   (b) *[15 marks]* Do you think P's comment is correct?

      i. If yes, indicate what subset of the inequalities represented in the octagon would suffice to define a new abstract domain. Describe the new abstract domain: the underlying set, ordering relation and why it is a lattice, the *lub* and *glb* operators, and a *widen* operator. Give also the abstraction and concretization operators, and show that they form a Galois connection.

      ii. If no, explain why it is not possible to obtain the exact count of iterations using a strict subset of the inequalities tracked by the octagon domain.