CS615 Autumn 2006 Compensation Quiz 7

Time: 30 mins Total marks: 10

- ***The exam is open book and notes.***
- ***Results/proofs covered in class/problem sessions/assignments may simply be cited, unless specifically asked for.***
- ***Unnecessarily lengthy solutions will be penalized.***
- ***Do not copy solutions from others or indulge in unfair means.***

1. Let the concrete domain be the set of subsets of all points in the real plane, i.e. all subsets of $\{(x, y) \mid x, y \in \Re\}$. Let the abstract domain be the set of all $6-$tuples of real numbers (a, b, c, d, e, f). We define the abstraction and concretization functions as follows:

   - If S is a set of points in the real plane, then $\alpha(S) = (a_S, b_S, c_S, d_S, e_S, f_S)$, where $a_S = \min(x)$, $b_S = \max(x)$, $c_S = \min(y)$, $d_S = \max(y)$, $e_S = \min(x - y)$ and $f_s = \max(x - y)$, the maximizations and minimizations being done over all points (x, y) in set S.
   - $\gamma((a, b, c, d, e, f)) = \{(x, y) \mid (a \leq x \leq b) \wedge (c \leq y \leq d) \wedge (e \leq x - y \leq f)\}$.

   We define the widening operator in the abstract domain as follows: $(a_1, b_1, c_1, d_1, e_1, f_1) \nabla (a_2, b_2, c_2, d_2, e_2, f_2) = (a_3, b_3, c_3, d_3, e_3, f_3)$, where

   - $a_3 = a_1$ if $a_2 \geq a_1$, and $-\infty$ otherwise
   - $b_3 = b_1$ if $b_2 \leq b_1$, and $+\infty$ otherwise
   - $c_3 = c_1$ if $c_2 \geq c_1$, and $-\infty$ otherwise
   - $d_3 = d_1$ if $d_2 \leq d_1$, and $+\infty$ otherwise
   - $e_3 = e_1$ if $e_2 \geq e_1$, and $-\infty$ otherwise
   - $f_3 = f_1$ if $f_2 \leq f_1$, and $+\infty$ otherwise

   Consider the following program `P` with location labels `Li`:

```
L1: while ((x-y >= 0) && (x-y <= 1) && (x <= 100)) {
L2:    x := x + y;
L3:    y := y + 1;
L4: }
```

   Suppose we start the program with the precondition $(0 \leq x \leq 1) \wedge (0 \leq y \leq 1)$. Compute as good a loop invariant (invariant at location `L1`) as possible by executing the program in the abstract domain and using the widening operator defined above. Your loop invariant should be given as an element of the abstract domain, i.e. as a $6-$tuple.