CS615 Graded Homework #2

Max Marks: 85 **Due date: Apr 9, 2010 (5 pm)**

- *Be brief, complete and stick to what has been asked.*
- *If needed, you may cite results/proofs covered in class without reproducing them.*
- ***Discussion among students is fine, but the solution you turn in must be your own solution in your own words. Cases of copying or indulgence in unfair means will be severely penalized, including award of FR grade.***

Problems on numerical abstract domains:

1. This exercise is a simple one, intended to illustrate the capabilities of different numerical abstract domains in obtaining location invariants in a program. Let P be the following program:

```
L1: while (i + j < 100)
L2:    i := j + 2;
L3:    j := j + 3;
L4: // end of while
```

   Assuming that the precondition of the program is $(i = 0) \wedge (j = 0)$, we wish to check whether the assertion $i = j+1$ holds at location `L4`. We will do this by first computing an inductive loop invariant at location `L2`. Compute a loop invariant at `L2` using:

   (a) *[10 marks]* the box (or interval) abstract domain, where an abstract state for our program is represented by a pair of intervals $([i_{low}, i_{high}], [j_{low}, j_{high}])$.

   (b) *[10 marks]* the DBM (difference bounds matrix) abstract domain, where an abstract state for our program is represented by a 3×3 matrix, as discussed in class.

   (c) *[10 marks]* the polyhedra abstract domain, where an abstract state for our program is represented by a set of linear constraints on i and j.

   In each case, use the *lub* operator to accumulate abstract states at `L1` for the first two iterations, and then apply widening until you obtain an inductive loop invariant. In each case, indicate whether the loop invariant computed by you suffices to prove the assertion $i = j + 1$ at location `L4`.

2. Let $(L, \sqsubseteq, \sqcup, \sqcap, \bot, \top)$ be a complete lattice with $\sqcup$ as the least upper bound operator and $\sqcap$ as the greatest lower bound operator. Let $l_1 \sqsubseteq l_2 \sqsubseteq \ldots \sqsubseteq l_k = \top$ be a chain of k elements (called *thresholds*) in the lattice, with the last element being the top element. For any two elements $x, y \in L$, we define $x \nabla_{th} y = l_i$, where i is the least index in $\{1, \ldots k\}$ such that $x \sqsubseteq l_i$ and $y \sqsubseteq l_i$.

   Prove the following:

   (a) *[5 marks]* ∇_{th} is a widen operator. Such a widen operator is also called a *threshold widen* operator for obvious reasons.

   (b) *[5 marks]* For all $x, y \in L$, $x \nabla_{th} (x \sqcup y) = x \nabla_{th} y$.

   (c) *[5 marks]* For all $x, y \in L$, $x \nabla_{th} (x \sqcap y) = x \nabla_{th} x$.

(d) *[5 marks]* You are told that $x \nabla_{th} (y \nabla_{th} z) = w \nabla_{th} w$. Express w in terms of x, y, z and the $\sqcap$ and $\sqcup$ operators, with justification.

3. In this question, we will reason about a program with two integer variables, `x` and `y`, using a new abstract domain. Let $\mathbf{Z}$ denote the set of integers and $\mathbf{Z}^+$ denote $\mathbf{Z} \cup \{-\infty, +\infty\}$. We will assume that $-\infty < +\infty$, $-\infty < n$ and $n < +\infty$ for all $n \in \mathbf{Z}$. Each element of our new abstract domain is a 4-tuple $(a, b, c, d) \in (\mathbf{Z}^+)^4$ such that either (i) $a \leq b$ and $c \leq d$, or (ii) $(a, b, c, d) = (+\infty, -\infty, +\infty, -\infty)$.

   The abstraction and concretization functions are defined as follows.

   - $\alpha : \wp(\mathbf{Z}^2) \rightarrow (\mathbf{Z}^+)^4$.
     For all non-empty $S \subseteq \mathbf{Z}^2$,
     $\alpha(S) = (\min_{(x,y)\in S}(x-y),\ \max_{(x,y)\in S}(x-y),\ \min_{(x,y)\in S}(x+y),\ \max_{(x,y)\in S}(x+y))$.
     If $S = \emptyset$, $\alpha(S) = (+\infty, -\infty, +\infty, -\infty)$.
   - $\gamma : (\mathbf{Z}^+)^4 \rightarrow \wp(\mathbf{Z}^2)$
     For all $(a, b, c, d) \in (\mathbf{Z}^+)^4$, if $a \leq b$ and $c \leq d$, $\gamma((a, b, c, d)) = \{(x, y) \in \mathbf{Z}^2 \mid (a \leq x - y \leq b) \wedge (c \leq x + y \leq d)\}$.
     $\gamma((+\infty, -\infty, +\infty, -\infty)) = \emptyset$.

   (a) *[3+3+4 marks]* Define suitable *lub*, *glb* and ∇ operators for our abstract domain, following ideas behind similar operators for the octagon domain studied in class.

   (b) *[5 + 5 + 5 marks]* Let (a, b, c, d) be an abstract domain element such that $a \leq b$ and $c \leq d$. Find the strongest abstract postcondition of (a, b, c, d) with respect to each of the following statements in our programming language.

   - `x := 2*x`
   - `x := 2 + x`
   - `y := y + x`

   In each case, your answer should be a four-tuple with each element of the tuple being an expression (in general) in a, b, c, d.

   (c) *[10 marks]* Now consider the following program in our programming language.

```
L0: while (x+y <= 100)
L1:   if (x+y <= 10) then
L2:     x := 2*x
L3:   else
L4:     x := 2 + x;
L5:   // end of if-then-else
L6:   y := y + x
L7: // end of while
```

   Assuming that the program is started in the abstract state $(0, 1, 0, 20)$, compute a loop invariant at `L0` using the following strategy.

   - In each iteration through the loop, compute the abstract state at `L5` as *lub* of the abstract post-conditions of the two brances of the `if then else` statement.
   - Compute the loop invariant at `L0` by taking the *lub* at the loop head after the first iteration of the loop, and then using the widen operator for all subsequent iterations.