Quiz 2

Synthesis, Analysis, and Verification 2010

Friday, May 27th, 2011

Last Name : ______________________________

First Name : ______________________________

Exercise	**Points**	**Achieved Points**
1	20	
2	40	
3	40	
Total	100	

Problem 1: Interval Analysis (20 points)

Consider interval analysis of a program with two integer variables x and y.
The state of the program is a map of the form $\{x \mapsto i, y \mapsto j\}$ where $i, j \in \mathbb{Z}$.
The abstract domain A associates an interval to each variable: it is the set of maps of the form

$$\{x \mapsto [l_x, u_x], y \mapsto [l_y, u_y]\}$$

with $l_x, l_y \in \mathbb{Z} \cup \{-\infty\}$ and $u_x, u_y \in \mathbb{Z} \cup \{\infty\}$.
The abstraction function α is such that given a set of concrete states S, $\alpha(S)(x)$ is the most precise interval containing all values of x found in S and $\alpha(S)(y)$ is the most precise interval containing all values of y found in S.
The concretization function γ is such that

$$\gamma(\{x \mapsto [l_x, u_x], y \mapsto [l_y, u_y]\}) = \{s.s(x) \in [l_x, u_x] \wedge s(y) \in [l_y, u_y]\}$$

Finally we define

$$sp^\sharp(a, c) = \alpha(sp(\gamma(a), c))$$

In the following questions the abstract postconditions need to be computed with respect to an arbitrary abstract precondition represented by $\{x \mapsto [l_x, u_x], y \mapsto [l_y, u_y]\}$. Please make sure that the abstract postconditions you give are as precise as possible.

Question 1.1. *Consider the statement* `y = 5*x^2 - 26*x +5`*. What is its abstract strongest postcondition?*

Question 1.2. *Give the abstract strongest postcondition for* `x = x*y` *and* `x = a*x + b*y`*.*

Question 1.3. *Use the rules you determined above to compute the abstract postcondition for the following program:*

```
y = 5*x - 1;
x = x - 5;
y = y*x;
```

Problem 2: Predicate Abstraction (40 points)

Consider a set of predicates

$$\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$$

Let us define an abstract domain A whose elements are sets of sets of predicates (A is the powerset of the powerset of $\mathcal{P}$).

Question 2.1. *Define a partial order $\sqsubseteq$ on A that does not rely on any interpretation of the predicates.*

Question 2.2. *Define a join operation $\sqcup$ such that $(A, \sqsubseteq, \sqcup)$ forms a semi-lattice.*

We interpret elements of the abstract domain as disjunctions of conjunctions:
Let γ_1 be a function mapping predicates to sets of concrete states, which are meant to represent the set of states for which the predicate is true.
Moreover given a set of predicates b we define

$$\gamma_2(b) = \bigcap_{p \in b} \gamma_1(p)$$

Finally the concretization function γ is such that given a set of set of predicates $a \in A$,

$$\gamma(a) = \bigcup_{b \in a} \gamma_2(b)$$

Question 2.3. *Make sure that γ is a monotonically increasing function (adapting your answer to the previous questions if not) and prove it.*

Consider the abstract strongest postcondition $sp^\sharp(a, c)$ where a is an element of the abstract domain A and c is a program command.
Given a set of sets of predicates $a \in A$, $sp^\sharp$ is defined as follows:

$$sp^\sharp(a, c) = \{sp_1^\sharp(b, c) | b \in a\}$$

where, given a set of predicates b, we have:

$$sp_1^\sharp(b, c) = \{p | p \in b \wedge \gamma_1(p) \subseteq wp(c, \gamma_1(p))\}$$

where wp is the weakest precondition operator.

Question 2.4. *Show that $sp^\sharp$ is sound.*

Now consider programs with two integer variables x and y and the following set of predicates:

$$P = \{0 \leq x, 0 \leq y, even(x), odd(y)\}$$

Assume that γ_1 is the obvious interpretation of the predicates.

Question 2.5. *Find a and c such that $\gamma(sp^\sharp(a, c))$ is a strict superset of $sp(\gamma(a), c)$.*

Finally, consider the following statement c:

```
x = y + 1
```

Question 2.6. *Is $sp^\sharp$ the most precise abstract postcondition for c? In other words, is there $a \in A$ for which there exists an element $a' \sqsubset sp^\sharp(a, c)$ such we still have:*

$$sp(\gamma(a), c) \subseteq \gamma(a')$$

Problem 3: Abstract Interpretation and Dynamic Memory Allocation (40 points)

Consider a language with the following grammar:

```
VAR  := Variable |
        VAR.next
EXPR := VAR |
        new Cell(EXPR, Nat) |
        null
CMD  := VAR = EXPR;
PROG := CMD PROG | EOF
```

where `Variable` is a set of variable names and `Nat` is the set of all natural numbers $\mathbb{N}$.
The language allows to dynamically allocate cells and to organize them in linked lists. The second parameter to the Cell constructor identifies allocation points in the program. We call it the tag of a cell. Here is an example of a program:

```
x = new Cell(null, 1)
y = new Cell(x, 2)
x.next = y
z = new Cell(y.next, 1)
```

Assume that there is a set of cells $C = \{C_i | i \in \mathbb{N}\}$ from which allocated cells are drawn. We denote the set of allocated cells by *Alloc*.
The state S of a program is defined as a triple $(pointsTo, next, Alloc)$ where $pointsTo : Variable \to Alloc \cup \{null\}$ and $next : Alloc \to \{Alloc \cup null\}$.
Initially no cells are allocated ($Alloc = \emptyset$). When new is used a cell is picked from $C \setminus Alloc$ and added to *Alloc*. Variable assignment updates the *pointsTo* component of the state. Finally initializations and updates of `next` in the program are reflected by initializations and updates of the *next* component of the state.

Question 3.1. *Describe a possible state after the last line of the example program has been executed.*

Let us now define an abstraction denoting, for any tag n, the set of tags that cells with tag n may point to and, for any variable, the set of tags the variable may point to. For this, define an abstract domain $S^\sharp$ whose elements are couples of partial functions $(pointsTo^\sharp, next^\sharp)$ where $pointsTo^\sharp : Variable \to 2^{\mathbb{N}}$ and $next^\sharp : \mathbb{N} \to 2^{\mathbb{N}}$.

Question 3.2. *Describe the abstract state corresponding to the state you determined in question 3.1*

Question 3.3. *Define an ordering relation $\sqsubseteq$ on the abstract domain such that $(S^\sharp, \sqsubseteq)$ is a partial order. Justify your answer by proving the relevant properties of $\sqsubseteq$.*

Question 3.4. *Define the join $\sqcup$ of two abstract states such that $(S^\sharp, \sqsubseteq, \sqcup)$ forms a semi-lattice. Justify your answer by proving the relevant properties of join.*

Question 3.5. *Define a concretization function γ that maps abstract states to sets of concrete states and that is meaningful for the analysis of programs in the language.*

Question 3.6. *For the statement* `x = y.next`*, prove that there exists the strongest abstract postcondition that satisfies the soundness requirement for abstract postconditions. Also describe how to compute it.*