CS615 Midsem Exam (Spring 2019)

Max marks: 60 **Time: 120 mins**

- *Be brief, complete and stick to what has been asked.*
- *Unless asked for explicitly, you can cite results/proofs covered in class.*
- ***If you need to make any assumptions, state them clearly.***
- ***Read the question paper carefully before answering questions.***
- ***Do not copy solutions from others. Penalty for offenders: FR grade.***

1. *[5 + 5 marks]* We have studied the notion of *weakest pre-condition* (WP) in class. Let P_a and P_b denote the code fragments (in a C-like language) shown below. Let φ denote the assertion `(x = y)`. Give expressions for $WP(\varphi, P_a)$ and $WP(\varphi, P_b)$. Answers without steps shown clearly will fetch zero marks.

```
int x, y;

if (x > 270) {
  x = y * y;
}
else {
  x = x * x;
  y = y * y;
}
```

Program P_a

```
linkedList *x, *y;

if ((x->next == NULL) && (y->next == NULL)) {
  y = x;
}
else {
  if (x->next != NULL) {
    x = x->next;
  }
  y = y->next;
}
```

Program P_b

2. *[10 + 10 marks]* We have studied about *Hoare triples* in class. In this question, we will explore different inference rules in a proof system using Hoare triples.

   We will restrict our attention to programs manipulating integers (no pointers) in the simple programming language studied in class. For each of the inference rules given below, either give an argument why the inference rule is sound, or provide a counterexample. If you are providing a counterexample, please provide one in our programming language and that does not use pointers.

   (a) Rule 1: $$\frac{\{\mathsf{True}\}\ P_2\ \{\psi_1\} \quad \{\mathsf{True}\}\ P_4\ \{\psi_3\} \quad \{\varphi_1\}\ P_1; P_2\ \{\psi_1 \Rightarrow \psi_2\} \quad \{\psi_2\}\ P_3; P_4\ \{\psi_3 \Rightarrow \psi_4\}}{\{\varphi_1\}\ P_1; P_2; P_3; P_4\ \{\psi_4\}}$$

   (b) Rule 2: $$\frac{\varphi_0 = \mathsf{True} \quad N \geq 1 \quad \forall j \in \{1, \ldots n\} \left(\{\bigwedge_{k=1}^{j-1} \varphi_{j-1}\}\ \texttt{P}\ \{\varphi_j\}\right)}{\{\mathsf{True}\}\ \texttt{int i=1; while (i <= N) do \{P; i=i+1;\} end while}\ \{\bigwedge_{1 \leq j \leq n} \varphi_j\}}$$

   In this problem, you may assume that the sub-program P doesn't modify the variable `i`.

3. *[10 + 10 marks]* In this question, we will try to develop a parameterized numerical abstract domain borrowing ideas from some numerical abstract domains studied in class. To keep things simple, we will assume that there are two `int` variables x_1, x_2 in the program that we are trying to analyze. Each element in the abstract domain is a pair of matrices (A, B), where $A_{k \times 2} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ \vdots & \vdots \\ a_{k1} & ak2 \end{bmatrix}$ and $B_{2 \times 1} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{bmatrix}$. The a_{ij}'s are all assumed to be real numbers. The b_i's can be real numbers or $+\infty$ or $-\infty$.

Note that the parameter $k(\geq 1)$ is *fixed* for the abstract domain, and is the same for every element of the abstract domain. There are some similarities between this abstract domain and the polyhedral abstract domain we studied in class. Specifically,

- The concretization function is exactly as in the case of polyhedra, i.e. $\gamma((A,B)) = \{(x_1,x_2) \mid A \cdot X \leq B\}$, where $X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$.
- Given (A_1,B_1) and (A_2,B_2), the linear programming based inclusion decision procedure for polyhedra can be used to decide if $(A_1,B_1) \sqsubseteq (A_2,B_2)$.

However, some other operators need to be designed differently, always over-approximating, if needed. In the following sub-questions, assume that (A_1,B_1) and (A_2,B_2) are arbitray elements of our parameterized (parameter k) abstract domain.

(a) We wish to use linear programming to design the most precise $\sqcup$ operator such that if $(A_3,B_3) = (A_1,B_1) \sqcup (A_2,B_2)$, then $A_3 = A_1$ and $\gamma((A_1,B_1)) \cup \gamma((A_2,B_2)) \subseteq \gamma((A_3,B_3))$.

Each linear program in our context can be specified by three real-valued matrices: $C_{l\times 2}, D_{1\times 2}, E_{1\times 2}$. Specifically, a linear program specified by these three matrices seeks the maximum value of $E \cdot X$ subject to the constraint that $C \cdot X \leq D$. Note that l may not be the same as k in general.

Describe how the elements of B_3 can be determined using at most k linear programs. You are required to give the C, D and E matrices for the (at most) k linear programs, and also indicate how the elements of B_3 are determined by the answers to these linear programs.

(b) Repeat the above sub-question, where we now wish to design the most precise $\sqcap$ operator such that if $(A_4,B_4) = (A_1,B_1) \sqcap (A_2,B_2)$, then $A_4 = A_2$ and $\gamma((A_1,B_1)) \cap \gamma((A_2,B_2)) \subseteq \gamma((A_4,B_4))$.

4. *[10 marks]* Consider the following program in a C-like programming language

```
L1: int i, j, k;

L2: i = j + k;
L3: if (i < k) then
L4:   while (i < k) do
L5:     i = i - (j/2);
L6:  end while
L7: else
L8:   while (i >= k) do
L9:     i = i - (j/20);
L10:  end while
L11: end if
```

Construct a Boolean program for the above program using the following predicates:

- $p_1 \equiv (j < 0)$
- $p_2 \equiv (i < k)$
- $p_3 \equiv (i > (j/2))$

Show clearly the steps in your derivation of the Boolean program.