Introduction to Software Verification 236342, Homework 1

Yosef Goren, Andrew Elashkin

November 2022

Question 1

a

For the program P prove the following using Floyd's proof system:

$$\{x = X \wedge y = Y\} P \left\{z = X^Y\right\}$$

We choose 3 cutpoints in the program: l_0, l_2 and l_*. The inductive assertions for those points are:

$$I_{l_0} = q_1 = (x = X \wedge y = Y)$$
$$I_{l_2} = (X^Y = z * x^y)$$
$$I_{l_*} = q_2 = (z = X^Y)$$

Now, For every basic path $\alpha = (l, l')$ we need to prove that for $\forall x$:

$$q_1(\bar{x}) \wedge R_{l_0 l_2}(\bar{x}) \rightarrow I\left(T_{l_0 l_2}(\bar{x})\right) \tag{1}$$
$$I(\bar{x}) \wedge R_{l_2 l_2}(\bar{x}) \rightarrow I\left(T_{l_2 l_2}(\bar{x})\right) \tag{2}$$
$$I(\bar{x}) \wedge R_{l_2 l_*}(\bar{x}) \rightarrow q_2\left(T_{l_2 l_*}(\bar{x})\right) \tag{3}$$

With this split we have 4 basic paths in our program flowchart, let us go through all of them one by one.

1. path $\alpha_1 : l_0 \rightarrow l_1 \rightarrow l_2$
   $R^2_\alpha(x, y, z) = \text{true} \rightarrow R^1_\alpha(x, y, z) = \text{true} \rightarrow R_\alpha(x, y, z) = \text{true}$
   $T^2_\alpha(x, y, z) = (x, y, z) \rightarrow T^1_a(x, y, z) = (x, y, 1) \rightarrow T_\alpha(x, y, z) = (x, y, 1)$

   Now we check the inductive assertion for α_1:
   $\forall(x, y, z)\left[((x = X \wedge y = Y) \wedge \text{true}) \rightarrow \left(X^Y = z * x^y\right)[(x, y, z) \leftarrow (x, y, 1)]\right] \equiv$
   $\equiv \forall(x, y, z)\left[((x = X \wedge y = Y) \wedge \text{true}) \rightarrow \left(X^Y = x^y\right)\right]$ and it holds.

2. path $\alpha_2 : l_2 \to l_*$
   $R^1_{\alpha_2}(x,y,z) = \text{true} \to R_{\alpha_2}(x,y,z) = (y=0)$
   $T^1_{\alpha_2}(x,y,z) = (x,y,z) \to T_{\alpha_2}(x,y,z) = (x,y,z)$

   Now we check the inductive assertion α_2:
   $\forall(x,y,z)\left[\left(\left(X^Y = z*x^y\right) \wedge (y=0)\right) \to \left(z = X^Y\right)[(x,y,z) \leftarrow (x,y,z)]\right] \equiv$
   $\equiv \forall(x,y,z)\left[\left(X^Y = z*x^0\right) \to \left(X^Y = z\right)\right]$ and it holds.

3. path $\alpha_3 : l_2 \to l_3 \to l_4 \to l_2$
   $R^3_\alpha(x,y,z) = \text{true} \to R^2_\alpha(x,y,z) = \text{true} \to$
   $R^1_\alpha(x,y,z) = \neg\text{even}(y) \to R_\alpha(x,y,z) = \neg\text{even}(y) \wedge \neg(y=0)$

   $T^3_\alpha(x,y,z) = (x,y,z) \to T^2_\alpha(x,y,z) = (x, y-1, z*x) \to$
   $T^1_a(x,y,z) = (x,y-1,z*x) \to T_\alpha(x,y,z) = (x,y-1,z*x)$

   Now we check the inductive assertion for α_3:
   $\forall(x,y,z)\left[(X^Y = z*x^y) \wedge (\neg\text{even}(y) \wedge \neg(y=0)) \to \left(X^Y = z*x^y\right)[(x,y,z) \leftarrow (x,y-1,z*x)]\right] \equiv$
   $\forall(x,y,z)\left[(X^Y = z*x^y) \wedge (\neg\text{even}(y) \wedge \neg(y=0)) \to (X^Y = (z*x)*x^{y-1})\right]$
   and it holds since $(z*x)*x^{y-1} = z*x^y$.

4. path $\alpha_4 : l_2 \to l_3 \to l_5 \to l_2$
   $R^3_\alpha(x,y,z) = \text{true} \to R^2_\alpha(x,y,z) = \text{true} \to$
   $R^1_\alpha(x,y,z) = \text{even}(y) \to R_\alpha(x,y,z) = \text{even}(y) \wedge \neg(y=0)$

   $T^3_\alpha(x,y,z) = (x,y,z) \to T^2_\alpha(x,y,z) = (x*x, y/2, z) \to$
   $T^1_a(x,y,z) = (x*x, y/2, z) \to T_\alpha(x,y,z) = (x*x, y/2, z)$

   Now we check the inductive assertion for α_4:
   $\forall(x,y,z)\left[(X^Y = z*x^y) \wedge (\text{even}(y) \wedge \neg(y=0)) \to (X^Y = z*x^y)[(x,y,z) \leftarrow (x*x, y/2, z)]\right] \equiv$
   $\forall(x,y,z)\left[(X^Y = z*x^y) \wedge (\neg\text{even}(y) \wedge \neg(y=0)) \to (X^Y = z*(x^2)^{y/2})\right]$ and
   it holds since $z*(x^2)^{y/2} = z*x^y$.

b

For the program P prove the following using Floyd's proof system:

$$\langle x = X \wedge y = Y \wedge Y \geq 0\rangle P \langle z = X^Y\rangle$$

In this question we will use the same cutpoints, the same calculations for R_α and T_α that we had in (a) and a well founded set $(\mathbb{N}, <)$ over natural numbers. The new inductive assertions will be:

$$I_{l_0} = ((w = y + 2) \wedge (x = X \wedge y = Y))$$
$$I_{l_2} = (w = y + 1) \wedge (X^Y = z * x^y))$$
$$I_{l_*} = ((w = y) \wedge (z = X^Y))$$

(INIT)

$$\forall(x, y, z)[(x = X \wedge y = Y \wedge Y \geq 0) \rightarrow \exists w[(w = y + 2) \wedge (x = X \wedge y = Y)]]$$

(DEC)

$$\forall(x, y, z, w)\,[I_l(x, y, z, w) \wedge R_\alpha(x, y, z) \rightarrow \exists w'\,[w' < w \wedge I_{l'}\,(T_\alpha(x, y, z), w')]]$$

Now we need to check if those conditions hold for the paths from (a)

1. path $\alpha_1 : l_0 \rightarrow l_1 \rightarrow l_2$

   $\forall(x, y, z, w)[((w = y + 2) \wedge (x = X \wedge y = Y) \wedge \text{true} \rightarrow$
   $\exists w'[w' < w \wedge ((w' = y + 1) \wedge (X^Y = z * x^y)[(x, y, z) \leftarrow (x, y, 1)]] \equiv$
   $\equiv \forall(x, y, z)\,[((w = y + 2) \wedge (x = X \wedge y = Y) \wedge \text{true}\,) \rightarrow \big(X^Y = x^y\big)]$

2. path $\alpha_2 : l_2 \rightarrow l_*$

   $\forall(x, y, z, w)[((w = y + 1) \wedge (X^Y = z * x^y)) \wedge (\text{y=0}) \rightarrow$
   $\exists w'[w' < w \wedge ((w' = 0) \wedge (X^Y = z)[(x, y, z) \leftarrow (x, y, z)]] \equiv$
   $\equiv \forall(x, y, z)\,\big[((w = 1) \wedge (z = X^Y) \rightarrow \exists w'[w' < w \wedge ((w' = 0) \wedge \big(X^Y = z\big)]$

3. path $\alpha_3 : l_2 \rightarrow l_3 \rightarrow l_4 \rightarrow l_2$

   $\forall(x, y, z, w)[((w = y + 1) \wedge (X^Y = z * x^y)) \wedge (\neg\text{even}(y) \wedge \neg(y = 0)) \rightarrow$
   $\big(X^Y = z * x^y)[(x, y, z) \leftarrow (x, y - 1, z * x)]\big] \equiv$
   $\exists w'[w' < w \wedge ((w' = y + 1) \wedge (X^Y = z * x^y)[(x, y, z) \leftarrow (x, y - 1, z * x)]] \equiv$

   $\forall(x, y, z, w)[((w = y + 1) \wedge (X^Y = z * x^y)) \wedge (\neg\text{even}(y) \wedge \neg(y = 0)) \rightarrow$
   $\big(X^Y = z * x^y)[(x, y, z) \leftarrow (x, y - 1, z * x)]\big] \equiv$
   $\exists w'[w' < w \wedge ((w' = y + 1) \wedge (X^Y = x * z * x^{y-1})]$

   and since $x * z * x^{y-1} = z * x^y$ it is true.

4. path $\alpha_4 : l_2 \rightarrow l_3 \rightarrow l_5 \rightarrow l_2$

   $\forall(x, y, z, w)[((w = y + 1) \wedge (X^Y = z * x^y)) \wedge (\text{even}(y) \wedge \neg(y = 0)) \rightarrow$
   $\big(X^Y = z * x^y)[(x, y, z) \leftarrow (x, y - 1, z * x)]\big] \equiv$
   $\exists w'[w' < w \wedge ((w' = y + 1) \wedge (X^Y = z * x^y)[(x, y, z) \leftarrow (x * x, y/2, z)]] \equiv$

   $\forall(x, y, z, w)[((w = y + 1) \wedge (X^Y = z * x^y)) \wedge (\text{even}(y) \wedge \neg(y = 0)) \rightarrow$
   $\big(X^Y = z * x^y)[(x, y, z) \leftarrow (x, y - 1, z * x)]\big] \equiv$
   $\exists w'[w' < w \wedge ((w' = y/2 + 1) \wedge (X^Y = z * (x^2)^{y/2})]$

   and since $z * (x^2)^{y/2} = z * x^y$ it is true.

Question 2

2.a

Let $P \in FPG'$ (extended Flowchart Programming Languge - with interrupts). And let $CFG : FPL \longrightarrow [n] \times [n]^2$ be a function returning the control flow graph any program $P \in FPG$ (unmodified).

Define $CFG' : FPL' \longrightarrow [n] \times [n]^2$ on input (P, P_{int}) to be the graph $CFG(P)$ with the following modifications: Any node $v \in P$ now has an additional output goes into the initial state of a private copy of $CFG(P_{int})$ (each $v \in P$ has it's own copy); Denote this private copy with S_v. Additionally, all final states of S_v go into a new node F_v, which goes back into v.

Define a reachability condition R and state transformation T for any path τ of FPL' program graphs as follows:

- Denote $\tau = l_{i_0}, l_{i_1}, ..., l_{i_k}$.
- Denote the state transformation of P_{int} (from it's initial to final state - or the one appended to it to be exact) with T_{int}.
- Let:
$$Option_{int}(S) = S_\tau(\bar{x}) \cup \{T_{int}(s) \mid s \in S_\tau(\bar{x}) \wedge q_{int}(s)\}$$
Inituitively, this transformation expands the set of states possible by possibly applying the state transformation of P_{int} to all states.
- Base case:
$$T^k_\tau(\bar{x}) = \{(\bar{x})\}, R^k_\tau(\bar{x}) = true$$
Note how now the transformation function outputs a set rather than a single state. This is meant to capture the indetermenism of the interrupt; it represents the set of all possible states that can result from an initial (input) state.
- Inductive case: Given the functions have been defined for $l_{i_{m+1}}$, define them over what is at label l_{i_m}:
  - *start* or *end*:
$$T^m_\tau(\bar{x}) = Option_{int}(T^{m+1}_\tau(\bar{x}))$$
$$R^m_\tau(\bar{x}) = R^{m+1}_\tau(\bar{x})$$
  - $\bar{x} := \bar{y}$:
    WLOG we can assume all assignments are to all variables - when this is not the case, we can append identity assignments.
$$T^m_\tau(\bar{x}) = Option_{int}(T^{m+1}_\tau(\bar{y}))$$
$$R^m_\tau(\bar{x}) = R^{m+1}_\tau(\bar{y})$$

– $B(\bar{x})$ (boolean branch expression):

$$T_\tau^m(\bar{x}) = Option_{int}(T_\tau^{m+1}(\bar{x}))$$

$$R_\tau^m(\bar{x}) = \begin{cases} R_\tau^{m+1}(\bar{x}) \wedge [\exists s \in B(T_\tau^{m+1}(\bar{x}))] & \text{if } l_{i_m} \rightarrow^T l_{i_{m+1}} \\ R_\tau^{m+1}(\bar{x}) \wedge [\exists s \in \neg B(T_\tau^{m+1}(\bar{x}))] & \text{if } l_{i_m} \rightarrow^F l_{i_{m+1}} \end{cases}$$

Inituitively; we require to get to the conditional label, and then have the boolean condition satisfiable by one of the possible states.

Now we define a modified floyed proof system which follows the steps on input (P, P_{int}):

1. Choose a set of cut points s.t.:
   - The set contains all initial and final states.
   - Every cycle in the graph $CFG'(P, P_{int})$ contains at least one cut point.
   - For every cut point l find an inductive assertion $I_l(\bar{x})$. Additionally it is required that: $I_{l_0}(\bar{x}) = q_1(\bar{x}), I_{l_*}(\bar{x}) = q_2(\bar{x})$ where l_0 is the initial state and l_* is a terminal state.
   - For every simple path between two cut points l_{i_m}, l_{i_j},

$$[I_{l_{i_m}}(\bar{x}) \wedge R_\tau^m(\bar{x}) \rightarrow I_{l_{i_j}}(\bar{x})]$$

The proof system is sound similarly to the original one; after the conditions have been shown - we know that for any path from l_0 to l_*, $I_{l_0}(\bar{x}) \rightarrow I_{l_*}(\bar{x})$ from closure on transitivity of the cut points conditions.
Since the initial and final conditions are the same as $q_1(\bar{x})$ and $q_1(\bar{x})$ - we have '$\{q_1(\bar{x})\}(P, P_{int})q_1(\bar{x})$'.

The new proof system is also reasonably complete, as haveing an interrupt which an [*false*] predicate would mean our proof system is equivalent to the original one.

2.b

The new proof system we define will be identical to the original floyed's system, with the following modifications:

1. The set of cut points must be the set of all vertecies in the CFG.

2. The new set of conditions is:
   For any simple path τ from two cut points l_{i_u}, l_{i_v} the following formula is satisfied:

$$[I_{i_u}(\bar{x}) \wedge R_\tau^{i_u}(\bar{x}) \rightarrow \neg q_{int}(T_\tau^{i_u}(\bar{x})) \wedge I_{i_v}(\bar{x})]$$

3. It is still required that $I_{l_0} = q_1$ (the precondition), but it is no longer required that $I_{l_*} = q_2$ (the postcondition).

The soundness of the proof system is due to the fact that the closure proprety will yield every cut point (thus every label) that is reachable to not satisfy the interrupt condition - meaning it is never possible to enter a run within any sate that can actually be reached.

It is also reasonably complete since we have to require every reachable state to not satisfy the interrupt condition.

Question 3

Let a *'critical'* be a point from which every proceeding point satisfies q_2.

The idea is to construct two sets of formulas which will each apply to the entier CFG, one will ensure that a *'critical'* point will be reached in the future (this can be done using well founded sets as seen in the lecture), and other will ensure that if the *'critical'* point is reached - the current point satisfies q_2 and the next point is also *'critical'*.

Formally, the proof system will require the following steps:

1. Define all vertecies in the CFG to be cut points.
2. Choose a well founded set $(W, <)$.
3. For each cut point l, define two formulas $Pre_l(\bar{x}, w)$ and $Post_l(\bar{x})$ where the first one depends on the state $\bar{x}$ and an item w from the well founded set - while the second only depends on the state.
4. Require this ('initial') formula to be satisfied:

$$\forall \left[q_1(\bar{x}) \rightarrow \exists w : Pre_{l_0}(\bar{x}, w)\right]$$

5. At every simple path τ between cut points l_{i_u}, l_{i_v}, require (all) the following formulas be satisfied:

   (a)

$$\forall \bar{x}, w \left[Pre_{l_{i_u}}((\bar{x}, w) \wedge R_\tau(\bar{x})) \rightarrow ((\exists w' < w : Pre_{l_{i_v}}(\bar{x}, w')) \vee Post_{l_{i_v}}(\bar{x}))\right]$$

   (b)

$$\left[Post_{l_u}(\bar{x}) \rightarrow Post_{l_v}(\bar{x})\right]$$

   (c)

$$\left[Post_{l_v}(\bar{x}) \rightarrow q_2(\bar{x})\right]$$

   Here in (a) we ensure that from any reachable point, we advance to a 'lower' point in the well founded set while maintaiing our set of formulas - or otherwise - reach the *'critical'* point.
   In (b) we ensure that once we reach the *'critical'* point, we maintain the *'critical'* property.
   Finally, (c) ties the critical property to the wanted *'postcondition'* q_2.

Here we have a sound proof system, since we have ensured that:

1. If the precondition is met - we will reach a *'critical'* point.
2. If we reach a *'critical'* point - we will maintain the *'critical'* property.
3. If we are passed a *'critical'* point - we satisfy the *'postcondition'* q_2.

If we wanted to prove this formally, we could follow a similar proof to the one shown in class of the proof system for showing termination properties using well founded sets.