

**Assignment 3: Program with Loops**  
**15-414/15-424 Bug Catching: Automated Program Verification**

Due: **11:59pm**, Thursday 9/21/17

Total Points: 50

1. **Prove the gcd's loop (10 points)** Give a sequent calculus proof, using the axioms of dynamic logic, that the following formula is valid.

$$0 < a \wedge 0 < b \rightarrow [c := a; d := b; \text{while}(c \neq d) \{ \text{if}(c > d) c := c - d \text{ else } d := d - c \}] (c = \text{gcd}(a, b))$$

You should assume that  $\text{gcd}$  is the standard greatest common divisor function. *Hint: you may find the following lemma useful, and are free to use it in your proof.*

$$\forall a, b \text{ gcd}(a, b) = \text{gcd}(a, b - a)$$

*Your loop invariant will also benefit from relating the gcd of the input variables to the gcd of the modified variables.*

2. **Find the invariants (5 points)** Provide loop invariants  $P$  and  $Q$  that are sufficient to prove validity of the following dynamic logic formula. You do not need to formally prove that the formula is valid, but you should succinctly explain why your invariants are correct.

$$\begin{aligned} n \geq 0 \rightarrow & [u := 1; \\ & r := 0; \\ & \text{while}(r < n) \text{invariant}(P) \{ \\ & \quad v := u; \\ & \quad s := 1; \\ & \quad \text{while}(s \leq r) \text{invariant}(Q) \{ \\ & \quad \quad u := u + v; \\ & \quad \quad s := s + 1; \\ & \quad \quad \} \\ & \quad r := r + 1; \\ & \quad \} \\ & ](u = n!) \end{aligned}$$

You should assume that the factorial function is defined normally:

$$n! = \begin{cases} 1 & \text{if } n = 0 \\ (n - 1)! n & \text{otherwise} \end{cases}$$

3. **Proof power (10 points)** Provide a loop invariant  $J$  sufficient to prove the following dynamic logic formula. Briefly explain the intuitive reason for the invariant (1-2 sentences). Use the axioms of dynamic logic along with your invariant  $J$  to give a sequent calculus proof of its validity.

$$0 \leq m \rightarrow [r := 1; i := 0; \text{while}(i < m) \{ r := r * n; i := i + 1 \}] (r = n^m)$$

*Note that if you have difficulty formatting your proof as a single tree, you are, as always, welcome to break it into multiple trees. But be careful to reference the correct subtrees in the overall proof. Be sure to give unique labels such as ① and ② to subtrees, and clearly reference them in your overall proof.*

4. **Unfold the soundness (5 points)** The lecture notes showed that the following axiom is a derived axiom so can be proved from other sound axioms in sequent calculus:

$$([\text{unfold}]) \quad [\text{while}(Q) \alpha] P \leftrightarrow (Q \rightarrow [\alpha][\text{while}(Q) \alpha] P) \wedge (\neg Q \rightarrow P)$$

Since axiom [unfold] is a derived axiom, it also is a sound axiom since it only proves valid formulas by soundness of the axioms and proof rules used in its sequent calculus proof. An alternative way of establishing soundness is a mathematical proof directly from the semantics of dynamic logic. Your job in this question is to give such a direct semantical soundness proof justifying that the following formula is valid just from the semantics of dynamic logic:

$$[\mathbf{while}(Q)\alpha]P \leftrightarrow (Q \rightarrow [\alpha][\mathbf{while}(Q)\alpha]P) \wedge (\neg Q \rightarrow P)$$

5. **While is just another repetition (5 points)** Lecture 3 defined a semantics  $\llbracket \mathbf{while}(Q)\alpha \rrbracket$  for the while loop  $\mathbf{while}(Q)\alpha$ . Lecture 5 defined a semantics  $\llbracket \alpha^* \rrbracket$  for the nondeterministic repetition  $\alpha^*$  and then went on to claim the following equivalence of programs:

$$\mathbf{while}(Q)\alpha \equiv \{?Q;\alpha\}^*; ?\neg Q$$

Use the semantics of programs to show that both programs are indeed equivalent by showing that the semantics of the left hand side is equal to the semantics of the right hand side, so both have the same reachability relation:

$$\llbracket \mathbf{while}(Q)\alpha \rrbracket = \llbracket \{?Q;\alpha\}^*; ?\neg Q \rrbracket$$

That is, show that the following are always equivalent:

$$(\omega, \nu) \in \llbracket \mathbf{while}(Q)\alpha \rrbracket \text{ iff } (\omega, \nu) \in \llbracket \{?Q;\alpha\}^*; ?\neg Q \rrbracket$$

6. **Soundness of while invariants (10 points)** Loop invariants are the most important reasoning technique for while loops. To disambiguate, this question will call their proof rule wloop:

$$(\text{wloop}) \frac{\Gamma \vdash J, \Delta \quad J, Q \vdash [\alpha]J \quad J, \neg Q \vdash P}{\Gamma \vdash [\mathbf{while}(Q)\alpha]P, \Delta}$$

Since we will be using the wloop invariant proof rule for while loops a lot throughout the whole semester, it is crucial to make sure—once and for all—that the proof rule is sound. Otherwise we would be conducting all kinds of formal sequent calculus proofs that do not actually imply validity of their conclusion, which would be rather futile. So before things go wrong, this is your chance to prove that the wloop rule is sound, which you will then be able to remember forever.

Recall that the lecture notes have proved soundness of a related invariant rule for nondeterministic repetition  $\alpha^*$ , which are closely related to while loops:

$$(\text{loop}) \frac{\Gamma \vdash J, \Delta \quad J \vdash [\alpha]J \quad J \vdash P}{\Gamma \vdash [\alpha^*]P, \Delta}$$

Prove soundness of the loop invariant rule wloop for while loops.

Hint: you may want to benefit from the fact that the lecture notes showed the loop rule for nondeterministic repetitions to be a derived rule.

7. **While while invariants are unsound (5 points)** Consider the following modified formulation of the while loop rule (changes highlighted in bold):

$$(\text{R4}) \frac{\Gamma \vdash J, \Delta \quad J \vdash [\alpha]J \quad J \vdash P, \mathbf{\Delta}}{\Gamma \vdash [\alpha^*]P, \Delta}$$

Show that rule R4 is unsound. That is, give an instance of rule R4 with concrete formulas in which all premises are valid but the conclusion is not valid. Briefly explain why that happens. Where in your proof of Task 6 do you rule out this counterexample?