CS615 Quiz3 (Autumn 2016)

Max marks: 20+20(bonus) **Time: 1 hour**

1. Consider the following program P in a C-like language, in which all variables are of type `int`.

```
L1: while (a <= 1000) {
L2:    a = b + i;
L3:    b = a + 1;
L4:    i = i + 1;
L5: }
L6: if (b > 2000) {
error:    exit(-1);
L7: }
```

We wish to check if starting from the pre-condition $\{(a \leq 1000) \wedge (b = 0) \wedge (i = 0)\}$ the program P can reach the `error` location. We will try to do this using predicate abstraction, i.e. using the technique of Boolean programs.

(a) *[10 marks]* Suppose we start off with the predicates $(a \leq 1000), (b \geq 0), (i = 0)$, and $(b \leq 2000)$. Let the boolean variables p_1, p_2, p_3, and p_4 denote the above predicates respectively.
Construct a Boolean program BP_1 from P using the above predicates. You must clearly indicate how each statement of the Boolean program is derived.

(b) *[5 marks]* Show that BP_1 constructed above can indeed reach the `error` location, when started with the pre-condition $\{p_1 \wedge p_2 \wedge p_3\}$. Give the shortest trace τ (i.e., sequence of program locations) in BP_1 that illustrates how the `error` location is reached in BP_1.

(c) *[5 marks]* Show that the shortest trace τ obtained above does not correspond to an execution of the original program P_1. In other words, show that the counterexample provided by the Boolean program is *spurious*.

(d) *[10 marks]* Using the idea of Craig interpolants, derive a location-specific set of predicates from the above shortest trace τ, that would eliminate the spurious counterexample obtained above. In other words, you are required to identify one or more predicate(s) corresponding to each location of the program, such that if we construct a new Boolean program BP_2 using the new set of predicates, the trace τ would not correspond to an execution of BP_2.

(e) *[10 marks]* Construct BP_2, i.e. a Boolean program from P using the location-specific predicates identified in the previous sub-question.