Introduction to Software Verification - HW No. 1

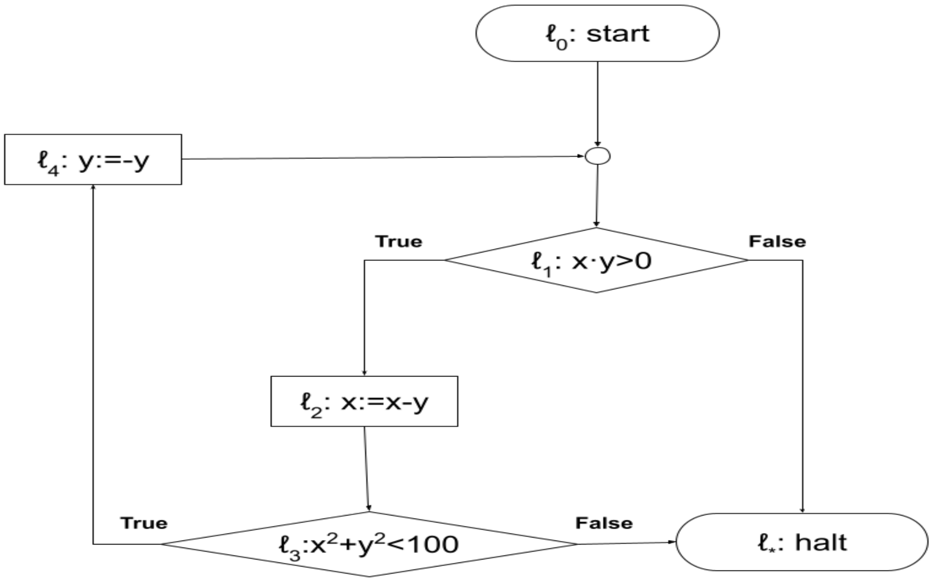
Winter 2022-2023

TA in charge of this HW: Roy Deutch

Pay attention: an answer without explanation will not be checked.

Question 1

Let P be the next program:
Which of the next specifications are correct?
Explain your answer (there is no need to prove it).



A. $\{false\}P\{x = 1 \wedge y = 1\}$
B. $\{true\}P\{y \geq 0 \vee x \geq 0\}$
C. $\{x = y\}P\{true\}$
D. $\{x < y \wedge x > 0 \wedge y < 10\}P\{false\}$
E. $\langle x^2 < y^2\rangle P \langle x^2 < y^2\rangle$
F. $\{x^2 < y^2\}P\{x^2 < y^2\}$
G. $\langle z = 5\rangle P \langle z = 5\rangle$
H. $\{z = 5\}P\{z = 5\}$
I. $\{z = y \wedge x > 0\}P\{z^2 = y^2\}$
J. $\langle x \geq y \wedge y > 0 \wedge x < 8\rangle P \langle x^2 + y^2 < 100\rangle$
K. $\langle x^2 - 2xy + 2y^2 < 10 \wedge x > 0 \wedge y > x\rangle P \langle true\rangle$
L. $\{x > y \wedge x < 0 \wedge x^2 + y^2 < 50\}P\{false\}$

Note: in questions 2,3 you can use the binary operators $+, \cdot$ and the relations $=, <, >$ only.
You **can't** assume the existence of predicates like $odd(x)$, $prime(x)$ etc.

Question 2

Let k be a non-negative integer.

Remainder: Fibonacci sequence is defined by the next recursive formula

$$F_n = F_{n-1} + F_{n-2}, \text{ when } F_0 = 0, F_1 = 1.$$

A. Write a specification for a program that gets a non-negative integer r, in the variable r returns the value 1 if $r > F_k$, the value 0 if $r = F_k$ and the value -1 if $r < F_k$.
B. Assume the existence of first-order logic formula $Fib_k(r)$, which is satisfied if and only if $r = F_k$. Use it, and write a specification for a program that gets a non-negative integer x, in the variable y returns the minimal multiple of F_k which is greater than x, also the value of x need to be preserved.

Question 3

Write a specification for the next programs:

A. A program that doesn't stop if the initial value of x is not a square of a prime number.
B. A program which stops if the initial value of x and the initial value of y are coprime (i.e. $gcd(x, y) = 1$).

Question 4

In this question we'll see how we can compute $R_\tau(\bar{x})$ and $T_\tau(\bar{x})$ which were defined in the first tutorial in forward computation.
Let $\tau = l_{i_0}, l_{i_1}, ..., l_{i_k}$ be a path in the length of $k + 1$ in the program, and let's define $R'^{m}_\tau(\bar{x})$ and $T'^{m}_\tau(\bar{x})$ as the reachability condition and state transformer for the **prefix** $l_{i_0}, ..., l_{i_m}$ of τ.
There for $R'^{k}_\tau(\bar{x}) = R_\tau(\bar{x})$ and $T'^{k}_\tau(\bar{x}) = T_\tau(\bar{x})$.
Show how to recursively compute $R'^{m}_\tau(\bar{x})$ and $T'^{m}_\tau(\bar{x})$ (In a similar way to the computation of $R'^{m}_\tau(\bar{x})$ and $T'^{m}_\tau(\bar{x})$ you saw in the tutorial).

Good luck!