Introduction to Software Verification – HW No. 2

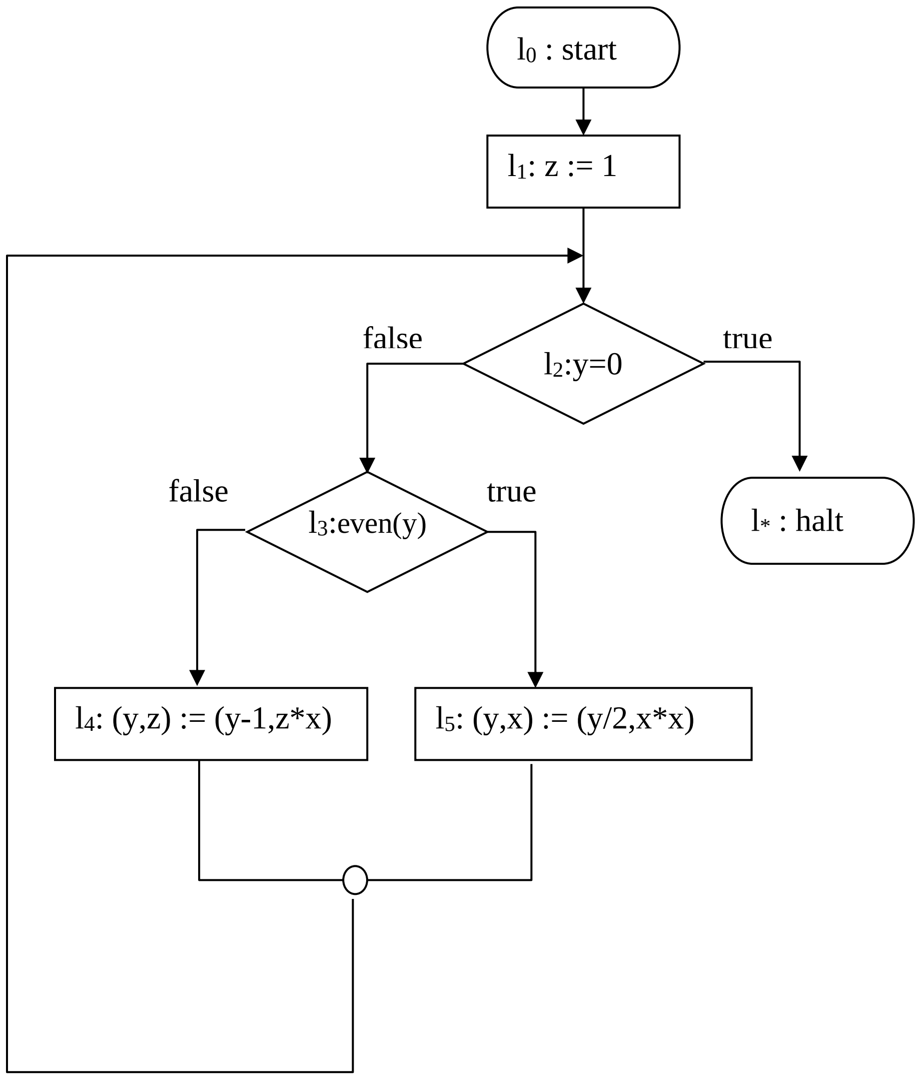
Winter 2022-2023

TA in charge: Omer Rappoport

Please note that answers without an explanation will not be checked.

Question 1

Let P be the following program:



a. Using Floyd's proof system, prove the following:

$$\{x = X \land y = Y\}P\{z = X^Y\}$$

b. Using Floyd's proof system, prove the following:

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

Assume that x and y are integers.

Question 2

We will extend the flowchart programming language to support interrupts:

Let P be a program and P_{int} be a program **without cycles**. Both programs operate on the same vector of variables $\overline{x}$. Also, let $q_{int}(\overline{x})$ be a precondition for executing P_{int}.

We say that the program P **is interrupted w.r.t. P_{int} and the condition q_{int}** if for every computation of P and for every state in which q_{int} holds, the program P_{int} is either executed or not non-deterministically. If P_{int} is executed, it runs fully until it halts (and it may change the values of the variables $\overline{x}$), and then the execution returns to the same point in the program P. It can be assumed that all commands in the flowchart programming language are atomic, i.e., P_{int} will not be executed while a command in P is executed. If q_{int} does not hold, the program P is guaranteed to continue as usual (without interrupts).

Let P be a program that is interrupted w.r.t. P_{int} and the condition q_{int}. If every terminating computation of P that starts from a state that satisfies q_1 ends in a state that satisfies q_2, we denote: $\{q_1\}P||P_{int}(q_{int})\{q_2\}$.

a. Let P be a program that is interrupted w.r.t. P_{int} and the condition q_{int}. Write a sound and complete proof rule, as much as possible, for proving $\{q_1\}P||P_{int}(q_{int})\{q_2\}$. Explain your answer.

b. Let P be a program that is interrupted w.r.t. P_{int} and the condition q_{int}. Write a sound and complete proof rule, as much as possible, that guarantees that P_{int} is never executed during an execution of P. Explain your answer.

Question 3

Let P be a program in the flowchart programming language, and let q_1, q_2 be first order logic formulas over the variables of the program. We denote $P \vDash q_1 \rightarrow EventuallyGlobally\ q_2$ if for every infinite execution of P, if it starts from a state that satisfies q_1 then there exists a state in the execution path such that from this state and on all states satisfy q_2. There is no requirement on finite executions.

Write a sound and complete proof rule, as much as possible, for proving $P \vDash q_1 \rightarrow EventuallyGlobally\ q_2$. Explain the soundness and completeness of the rule briefly.

Hint: Select all program labels as cut points.

Note: Do not assume that every execution of P that starts from a state that satisfies q_1 is finite.

Good luck!