

2G1516 Formal Methods – HT05

Floyd/Hoare Logic

Inductive Assertion Method

Exercise 1 Consider the transition diagram P on fig. 1 where all variables range over the non-zero natural numbers: Let ϕ_{pre} be the assertion $x \geq 1$ and ϕ_{post} be the assertion $u = x!$. Prove, using Floyd’s inductive assertion method that P is partially correct for ϕ_{pre} and ϕ_{post} . (Hint: For s_1 use the invariant assertion $y = \frac{x!}{z!} \wedge u = y$ and for s_2 use the invariant assertion $u = y(z - v + 1) \wedge y = \frac{x!}{z!}$.) \square

Exercise 2 Let variables x and y range over the natural numbers and let $P = \langle S, \Sigma, R, s_0, s_f \rangle$ be the transition diagram where:

- $S = \{s_0, s_f\}$
- $\Sigma = \{\alpha_1, \alpha_2, \alpha_3\}$ where

$$\begin{aligned} \alpha_1 &\stackrel{\text{def}}{=} x < y \rightarrow y := y - x \\ \alpha_2 &\stackrel{\text{def}}{=} y < x \rightarrow x := x - y \\ \alpha_3 &\stackrel{\text{def}}{=} x \equiv y \rightarrow skip \end{aligned}$$

- $R = \{(s_0, \alpha_1, s_0), (s_0, \alpha_2, s_0), (s_0, \alpha_3, s_f)\}$.

The definition corresponds to Dijkstra’s *greatest common divisor* (GDC) algorithm: at state s_f , both variables x and y contain $gcd(X_0, Y_0)$ where X_0 and Y_0 are the original values of variables x , resp. y .

1. Draw the transition diagram P .
2. Write the state space generated by the transition diagram P .
3. Express $gcd(x, y)$ using a FOL formula.
4. Give a partial correctness specification $\{\phi_{pre}\} P \{\phi_{post}\}$ that reflects the fact that P computes the greatest common divisor of the original values of x and y . (Should the cases when either x or y are initially zero be excluded from ϕ_{pre} ?).

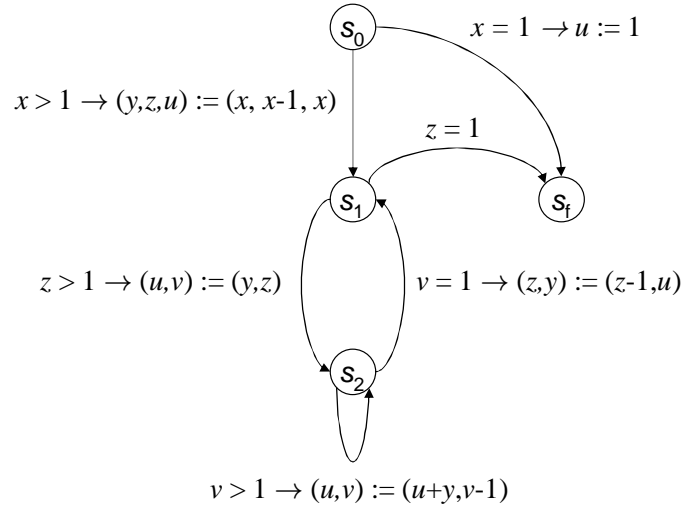


Figure 1: Transition diagram for exercise 1

5. Using Floyd's Inductive Assertions Method, show the partial correctness of P w.r.t. ϕ_{pre} and ϕ_{post} .
6. Define a well-founded order $(W, <)$ and an assignment w so that you can apply the Extended Inductive Assertion Method to prove that P is totally correct w.r.t. ϕ_{pre} and ϕ_{post} .

□

Exercise 3 (David Gries' Coffee Can Problem) Given a can of black and white coffee beans, repeat the following until there is only one bean left in the can:

Pull out two beans at random. If they are of the same color, replace them with a black bean (enough extra black beans are available to do this); otherwise, replace them with a white bean.

1. Model the problem as a transition diagram.
2. If you know how many beans of each color are in the can to begin with, can you predict the color of the final bean?
3. Verify your answer to the question above using Floyd's Extended Inductive Assertion Method.

□

Hoare Logic

Exercise 4 (Swapping with no temporary) Give a Hoare-logic proof of

$$\begin{aligned} & \{x = X \wedge y = Y\} \\ & x := x + y; \\ & y := x - y; \\ & x := x - y; \\ & \{y = X \wedge x = Y\} \end{aligned}$$

□

Exercise 5 (Maximum) Prove the following partial correctness result using Hoare-logic:

$$\begin{aligned} & \{true\} \\ & \mathbf{if} \ x \geq y \ \mathbf{then} \\ & \quad z := x \\ & \mathbf{else} \\ & \quad z := y \\ & \mathbf{fi} \\ & \{z = \max(x, y)\} \end{aligned}$$

□

Exercise 6 Write a program in the WHILE-language described in the lecture to compute the greatest common divisor of two positive integers. Use Hoare logic to show that the program satisfies the partial correctness specification given in Exercise 1.4. □

Exercise 7 (Quotient and Remainder) Prove using Hoare-logic that:

$$\begin{aligned} & \{n \geq 0 \wedge d > 0\} \\ & q := 0; \\ & r := n; \\ & \mathbf{while} \ r \geq d \ \mathbf{do} \\ & \quad q := q + 1; \\ & \quad r := r - d \\ & \mathbf{od} \\ & \{n \equiv q * d + r \wedge 0 \leq r < d\} \end{aligned}$$

□

Exercise 8 Prove carefully the following lemma using the operational semantics for while programs given in the lecture.

Lemma: If $(c_1; c_2, \sigma) \rightarrow^n \sigma'$ then there are n_1, n_2, σ'' such that $(c_1, \sigma) \rightarrow^{n_1} \sigma''$, $(c_2, \sigma'') \rightarrow^{n_2} \sigma'$, and $n = n_1 + n_2$.

Use this lemma to prove soundness of the Hoare Logic rule for **while**. □