Homework Assignment #0

Due Time/Date

2:20PM Wednesday, September 11, 2024. The purpose of this homework is for you to get warmed up and will not be counted as part of your grade of this course.

How to Submit

Please write or type your answers on A4 (or similar size) paper. Put your completed homework on the instructor's desk before the class on the due date starts. For early or late submissions, please drop them in Yih-Kuen Tsay's mail box on the first floor of Management College Building 2. You may discuss the problem with others, but copying answers is strictly forbidden.

Problems

1. Consider the "untangling line segments" problem discussed in class. Your task is to refine the rank function given in class so that it maps program states to *non-negative integers*. Try to be precise (using mathematical concepts and notations). You must show in sufficient details that the execution of an untangling operation, if enabled, will decrease the value of the rank function by *at least one*.

2. Below is a function implementing a variant of Euclid's algorithm:

   Function originalEuclid(m, n);
   begin
   // assume that $m > 0$ and $n > 0$
   $x := m$;
   $y := n$;
   while $x \neq y$ **do**
   if $x < y$ **then** swap(x,y);
   $x := x - y$;
   od
   ...
   end

   where swap(x,y) exchanges the values of x and y.

   (a) Please give a suitable loop invariant for the while loop. The loop invariant should be strong enough to determine that, when the program exits the while loop, the value

of x (or y) equals the greatest common divisor of m and n. Try to be as precise as possible.

(b) Please also give a suitable rank function (mapping program states to non-negative integers) for the while loop. Each iteration of the while loop (when the Boolean condition holds) should reduce the value of the rank function by at least one. Again, try to be as precise as possible.

3. Consider the following function for computing the square of the input number n, which is assumed to be a positive integer.

Function mySquare(n);
begin
    // assume that $n > 0$
    $x := n$;
    $y := 0$;
    while $x > 0$ **do**
        $y := y + 2 \times x - 1$;
        $x := x - 1$
    od;
    mySquare := y
end

Please give a suitable loop invariant for the while loop. The loop invariant should be strong enough for deducing that, when the while loop terminates, the value of y equals the square of n.