Eliot Piering 9-11-14

Homework4

1. Problem 1.

Table 1: Nim game table

Marbles	Strategy	Win or Lose
1	P1 has to take the last marble	lose
2	P1 takes 1 marble leaving p2 with 1 marble	win
3	p1 takes 2 marbles	win
4	p1 takes 3 marbles	win
5	p1 takes any amount	lose
6	p1 takes 1 leaving p2 with 5 marbles	win
7	p1 takes 2 marbles	win
8	p1 takes 3 marbles	win
9	p1 takes any amount	lose
10	p1 takes 1 marble leaving p2 with 9 marbles	win
11	p1 takes 2 marbles	win

Proposition: P(n): Player 1 will win if the amount of marbles is equal to 4n - 2, 4n - 1, or 4n for all n > 0

Proof:

Base Case: n = 0; P(1),

case 1, 4n - 2 = 2, P1 takes a marble and leaves P2 in the losing position of 1 marble

case 2, 4n - 1 = 3, P1 takes 2 marbles and leaves P2 in the losing position of 1 marble

case 3, 4n = 4, P1 takes 3 marbles and leaves P2 in the losing position of 1 marble

Induction Hypothesis:

P(n) $\longrightarrow$ P(n+1)

Proof:

Assume P(n) for all piles of size 4n -2, 4n -1, or 4n p1 wins

case 1, 4(n+1) - 2 == 4n + 4 -2 == 4n + 2 P1 can take away 1 marble leaving 4n + 1, now P2 has to take either 1,2,3 or 5 marbles which would leave 4n, 4n -1 or 4n - 2 marbles so P1 would be in a winning situation

case 2, 4(n+1) - 1 == 4n + 4 -1 == 4n + 1 P1 can take away 2 marbles leaving 4n + 1, now P2 has to take either 1,2,3 or 5 marbles which would leave 4n, 4n -1 or 4n - 2 marbles so P1 would be in a winning situation

case 3, 4(n+1) == 4n + 4 P1 can take away 3 marbles leaving 4n + 1, now P2 has to take either 1,2,3 or 5 marbles which would leave 4n, 4n -1 or 4n - 2 marbles so P1 would be in a winning situation

So by the induction hypothesis for all piles size 4n -2, 4n - 1, or 4n P1 will win.

2. Problem 2

   (a) { x > 0 } x := x +1 { x > 0 }
   If X is greater than 1 before the program begins, and then you add 1 to x, x will still be greater than 1 when the program ends.

   (b) { x > 0 } x := x -1 { x > 0 }
   This program is not correct. If x = 1 to start the program, then subtracting 1 in the program will leave x to be 0. 0 is not greater than 0 so the final condition will be false

(c) { $x < 0$ }

while $x \neq 0$ do

x := x + 1

od

{ $x = 0$ }

a = x

The loop invariant is $a \leq 0$.

If $a < 0$ after the first iteration of the loop a‘ = a + 1. a = a‘ - 1. So a‘ - 1 < 0, a‘ < 1; given a is an integer a‘ < 1 is the same as a‘ ≤ 0. So the loop invariant is true after 1 iteration of the loop.

Assume $a < 0$ and through all iterations of the loop $a \leq 0$ prove the postcondition $x = 0$ when the loop terminates. This loop will terminate only when $a = 0$, and because a is always ≤ 0 and each iteration through the loop the value of a moves toward 0, the loop will eventually terminate when $a = 0$, so the postcondition will be met.

Given $x \neq 0$ and

(d) { true }

while $x \neq 0$ do

x := x + 1

od

{ $x = 0$ }

This program is incorrect. If x starts off greater than 0 then this loop will never terminate. The loop will only terminate when $x = 0$, howerver if you continue adding 1 to x when x is already greater than 0, x will never equal 0. Therefore this program is incorrect.