Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D. Basin and P. Müller

Formal Methods and Functional Programming

Solutions of Exercise Sheet 7: The Programming Language **IMP**

Assignment 1

```
/* compute the floor of the x-th root of y */
z := 0;
v := 0;
while v < y do
    /* compute (z+1)^x and store it in the auxiliary variable v */
    v := 1;
    i := 0;
    while i < x do
        v := v * (z+1);
        i := i+1
    end;
    /* increase result when still below input value */
    if v <= y  then
        z := z+1
    end
end
```

We first give arguments that the program terminates whenever the variables x and y store in the initial state the values $m > 0$ and $n \geq 0$, respectively. For simplicity, we will not distinguish between a variable and the value that it stores when the state is clear from the context.

Let us observe that the inner while loop terminates. This follows from the assumption that initially $\mathtt{x} > 0$ and x is never assigned to a new value in the program. Note that in each iteration i increases by 1.

Now, we argue that the outer while loop terminates. We observe that y never gets assigned to a new value. So, since the inner while loop terminates in each iteration of the outer while loop, it suffices to show that in each iteration, the value stored in v after the inner while loop is greater than its value in the previous iteration at that program point. In fact, this is the case, since z gets increased at the end of each iteration if $\mathtt{v} \leq \mathtt{y}$. Hence, if $\mathtt{v} < \mathtt{y}$ at the end of the outer while loop, we have that in the next iteration at the end of the inner while loop, v stores a greater value. So, the guard of the outer while loop eventually becomes false.

Let us now argue that the program computes $\lfloor \sqrt[\mathtt{x}]{\mathtt{y}} \rfloor$ under the assumption that the program terminates. We first observe that the inner while loop satisfies the invariant $\mathtt{v} = (\mathtt{z}+1)^{\mathtt{i}}$. It follows that after the inner while loop, $\mathtt{v}$ stores the value $(\mathtt{z}+1)^{\mathtt{x}}$.

We claim that the outer while loop satisfies the invariant $\mathtt{v} \leq \mathtt{y} \rightarrow \mathtt{z}^{\mathtt{x}} \leq \mathtt{y}$. This follows from $\mathtt{v} = (\mathtt{z}+1)^{\mathtt{x}}$ after the inner while loop and the if statement at the end of the outer while loop. We conclude that after the outer while loop, we have that $\mathtt{z}^{\mathtt{x}} \leq \mathtt{y} \wedge \mathtt{v} \geq \mathtt{y}$. Since $\mathtt{v} = \mathtt{z}^{\mathtt{x}} \vee \mathtt{v} = (\mathtt{z}+1)^{\mathtt{x}}$ at the end of the while loop, it follows that $\mathtt{z} = \lfloor \sqrt[\mathtt{x}]{\mathtt{y}} \rfloor$.

Assignment 2

Let $n \geq 1$ be the number of cards in each pile.

1. Base case: $n = 1$. Player I can only remove 1 card from a pile. There is no other choice for him. Player II can remove the remaining card in the other pile. Since it is the last card, Player II wins.

2. Step case: $n > 1$. The induction hypothesis is that Player II can win the game when there are $1 \leq m < n$ cards on each pile.

   Assume that Player I removes $j > 0$ cards from a pile. Player II removes the same number of cards from the other pile. We have now that each pile has $n - j < n$ cards. If $n - j \geq 1$, then by induction hypothesis Player II can win from this game configuration. Thus, he can also win the game with n cards in each pile. If $n - j = 0$, then Player II has removed the last card, and therefore he wins the game.

When using weak induction the induction hypothesis is too weak to prove the induction step.