FPV 3 - FPV Summary for the week 3

Funktionale Programmierung (Technische Universität München)

It traverses the program till termination, if the program terminates.

2. Loop invariant: an assertion that holds in each iteration of the loop. Mostly, it the changing variable is often determined inside an interval instead of a definite value.

3.Termination:

a) A program without loops or recursive calls always terminates.

b) For loops where the indicator i, j, etc is not modified in the body always terminates.

c) In while loops, where there is no obvious indicator, we have to find an indicator from the given variables or generate one ourselves. If they are strict monotone in the function/method and the initial value can be determined explicitly, they can be used as an indicator.

2 Minijava 2.0

1. $WP[\![rand \quad x]\!](B) \equiv \forall x.B$

2. $WP[\![x = either \quad e_0, e_1, ..., e_n]\!](B) \equiv B[e_0/x] \wedge B[e_1/x] \wedge ... \wedge B[e_n/x]$

3. $WP[\![x = e \; in \; a, b]\!](B) \equiv$

$(a \leq x \leq b \implies B[1/x]) \wedge (e < a \vee e > b \implies B[0/x])$

4. $WP[\![stop]\!](B) \equiv true$

3 Loop Invariant

In this section, we are handling the WP in the scenarios of the loop invariants in WPs. As is discussed in the recap, we have to find an invariant concerning the indicator of the loop and other variables. In this case, an explicit i is given and no further search is necessary.

$\mathbf{WP}[\![write(x)]\!](x = n!) \equiv x = n! :\equiv A$

By analyzing the program we can find that what we compute in the loop is actually the factorial of n, so we assume $x = i!$ in the first place. Additionally, we might also have to determine the interval in which i locates.

Assume: $I :\equiv x = i! \wedge 0 < i \leq n$, then we shall prove our assumption.

$\mathbf{WP}[\![x = x * i]\!](x = i! \wedge 0 < i \leq n) \equiv x = (i-1)! \wedge 0 < i \leq n :\equiv B$

$\mathbf{WP}[\![i = i + 1]\!](x = (i-1)! \wedge 0 < i \leq n) \equiv x = i! \wedge 0 < i + 1 \leq n :\equiv C$

$\mathbf{WP}[\![i < n]\!](A, C) \equiv (i \geq n \wedge x = n!) \vee (i < n \wedge x = i! \wedge 0 < i + 1 \leq n) \equiv$

$(i \geq n \wedge x = n!) \vee (0 < i < n \wedge x = i!) \Longleftarrow (i = n \wedge x = n!) \vee (0 < i < n \wedge x = i!) \equiv$

$x = i! \wedge 0 \leq i \leq n :\equiv D$

So far, we have done with the loop invariant, and should continue with the rest of the control flow graph. Two open questions can be raised here: **How do we determine the formulae used in the loop invariant? What would happen if the interval of i is not given or incorrectly given?**

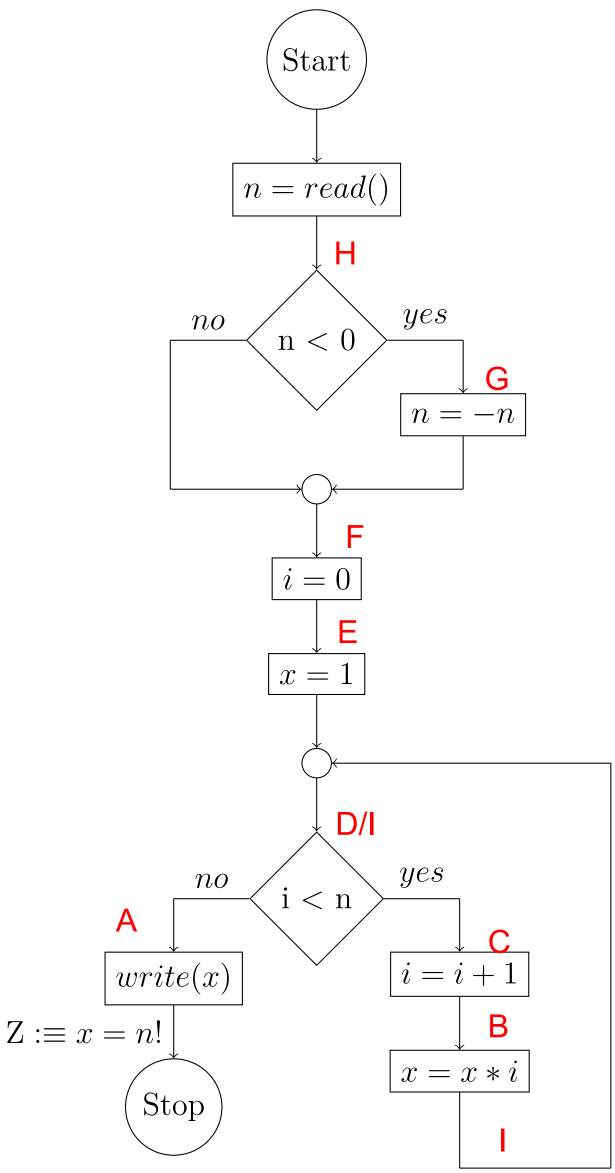
$\mathbf{WP}[\![x = 1]\!](D) \equiv i! = 1 \wedge 0 \leq i \leq n :\equiv E$

$\mathbf{WP}[\![i = 0]\!](E) \equiv i = 0 \wedge 0! = 1 \wedge 0 \leq 0 \leq n \equiv n \geq 0 :\equiv F$

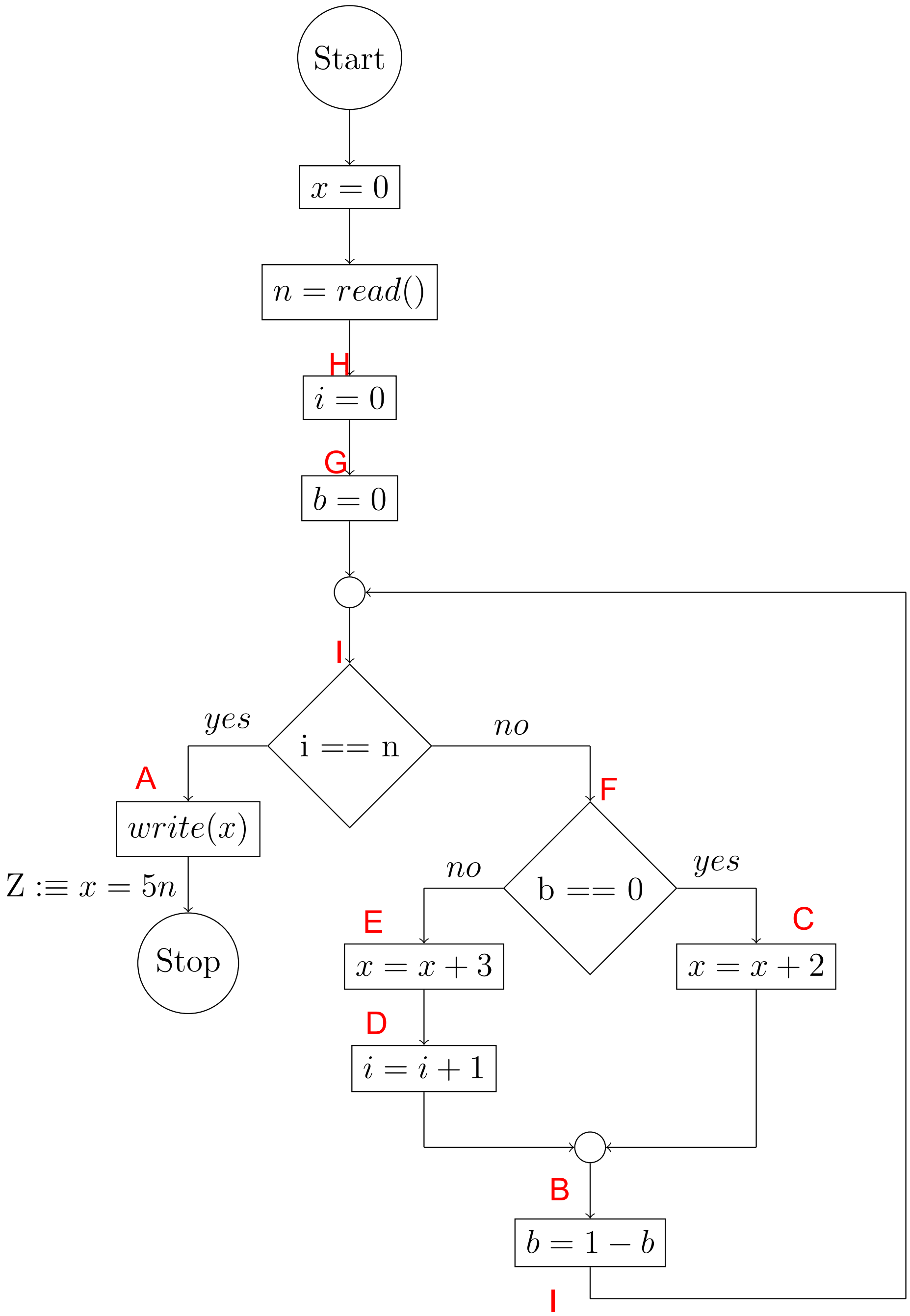
$\mathbf{WP}[\![n = -n]\!](F) \equiv n \leq 0 :\equiv G$

$\mathbf{WP}[\![n < 0]\!](F, G) \equiv (n \geq 0 \wedge n \geq 0) \vee (n < 0 \wedge n \leq 0) \equiv true :\equiv H$

$\mathbf{WP}[\![n = read()]\!](H) \equiv true$



4 Two b, or Not Two b



Now we are handling a more complicated case of loop invariant, the main idea does not actually change. We still can find an explicit indicator i, but finding the invariant is complicated due to the if-else statement inside the loop.

$\mathbf{WP}[\![write(x)]\!](Z) \equiv x = 5n :\equiv A$

We can tell from analysing the program that **b** is switching its value between 0 and 1 in each run of the loop. So intuitively we would add $b \in \{0, 1\}$ or simply $0 \leq b \leq 1$.

Then we find that what is computed here is 5 times of n, but inside the loop $x = 5i + 2 \lor x = 5i$, such value is determined by b. More exactly, we can find $(b = 0 \implies x = 5i) \land (b = 1 \implies x = 5i + 2)$.

Now what is left to do is the interval of i. In the same way, we intuitively assume i to be in the interval $(0, n]$ when getting out of the loop. But as we are not using the $<$ or $>$ as comparison. We don't care about the inequality, but only the state of the program when exiting the loop. Hence we assume $I \equiv (i = n \implies b = 0) \land b \in \{0, 1\} \land x = 5i + 2b$

$\mathbf{WP}[\![b = 1 - b]\!](I) \equiv (i = n \implies b = 1) \land b \in \{0, 1\} \land x = 5i + 2 - 2b :\equiv B$

$\mathbf{WP}[\![x = x + 2]\!](B) \equiv (i = n \implies b = 1) \land b \in \{0, 1\} \land x = 5i - 2b :\equiv C$

$\mathbf{WP}[\![i = i + 1]\!](B) \equiv (i + 1 = n \implies b = 1) \land b \in \{0, 1\} \land x = 5i + 7 - 2b :\equiv D$

$\mathbf{WP}[\![x = x + 3]\!](D) \equiv (i + 1 = n \implies b = 1) \land b \in \{0, 1\} \land x = 5i + 4 - 2b :\equiv E$

$\mathbf{WP}[\![b == 0]\!](E, C)$

$\equiv (b \neq 0 \land (i + 1 = n \implies b = 1) \land b \in \{0, 1\} \land x = 5i + 4 - 2b)$

$\lor (b = 0 \land (i = n \implies b = 1) \land b \in \{0, 1\} \land x = 5i - 2b) \equiv$

$(b = 1 \wedge (i + 1 = n \implies true) \wedge x = 5i + 2) \vee (b = 0 \wedge x = 5i - 2b \wedge i \neq n) \equiv$

$(b = 1 \wedge x = 5i + 2b) \vee (b = 0 \wedge x = 5i + 2b \wedge i \neq n) \Longleftarrow$

$b \in \{0, 1\} \wedge x = 5i + 2b \wedge i \neq n :\equiv F$

$\mathbf{WP}[\![i == n]\!](A, F) \equiv (i = n \wedge x = 5n) \vee (i \neq n \wedge x = 5i + 2b \wedge b \in \{0, 1\} \wedge i \neq n) \Longleftarrow$

$((i = n \implies b = 0) \wedge b \in \{0, 1\} \wedge i = n \wedge x = 5n) \vee ((i = n \implies b = 0) \wedge i \neq n \wedge b \in \{0, 1\} \wedge x = 5i + 2b) \equiv$

$(i = n \implies b = 0) \wedge x = 5i + 2b \wedge b \in \{0, 1\} \equiv I$

Again, we have proven our assumption, now we can continue with the control flow.

$\mathbf{WP}[\![b = 0]\!](I) \equiv x = 5i :\equiv G$

$\mathbf{WP}[\![i = 0]\!](G) \equiv x = 0 :\equiv H$

$\mathbf{WP}[\![n = read()]\!](H) \equiv x = 0 :\equiv J$

$\mathbf{WP}[\![x = 0]\!](J) \equiv true$