FPV Tutorübung

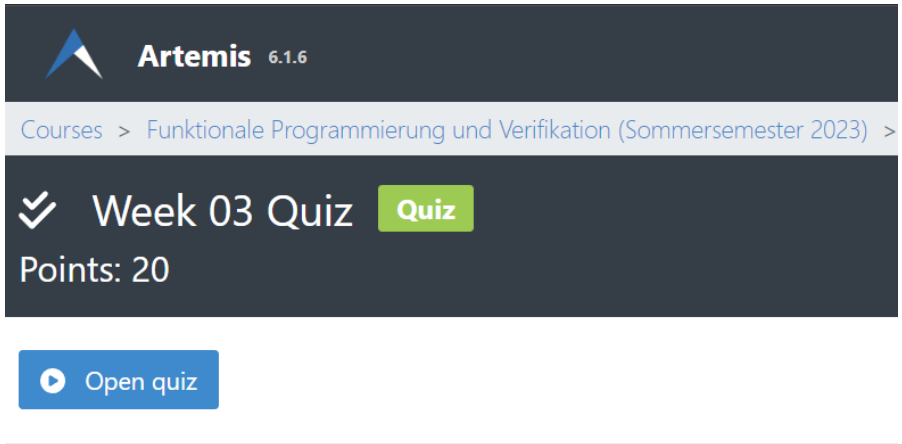
Woche 3

MiniJava 2.0, Loop Invariants

Manuel Lerchner

09.05.2023

Quiz



Passwort:

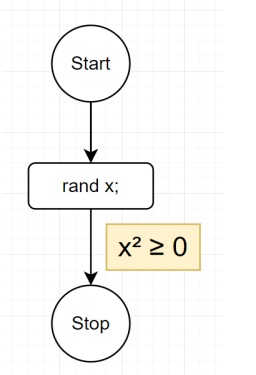
T01: MiniJava 2.0

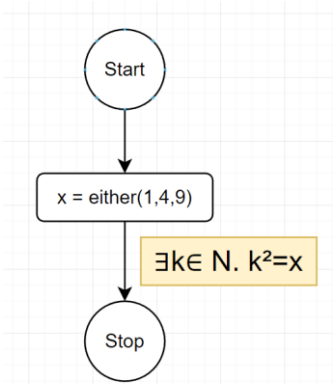
In the lecture, the weakest precondition operator has been defined for all statements of MiniJava. In this assignment, we consider an extension of the MiniJava language, which provides four new statements:

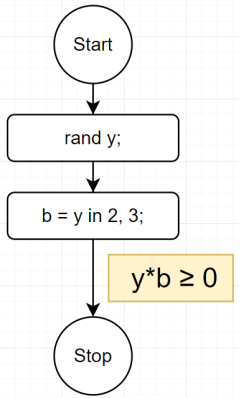
1. `rand x`:
   Assigns a random value to variable x,
2. `x = either` $e_0, \ldots, e_k$:
   Assigns one of the values of the expressions $e_0, \ldots, e_k$ to variable `x` non-deterministically,
3. `x = e in a, b`:
   Assigns the value 1 to variable `x`, if the value of expression `e` is in the range $[a, b]$ and 0 if `e` is not in the range or the range is empty $(a > b)$,
4. `stop`:
   Immediately stops the program.

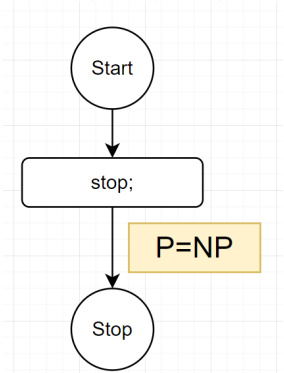
Define the weakest precondition operator $\mathbf{WP}[\![\ldots]\!](B)$ for each of these statements. (In terms of B)

Beispiele zum Testen:



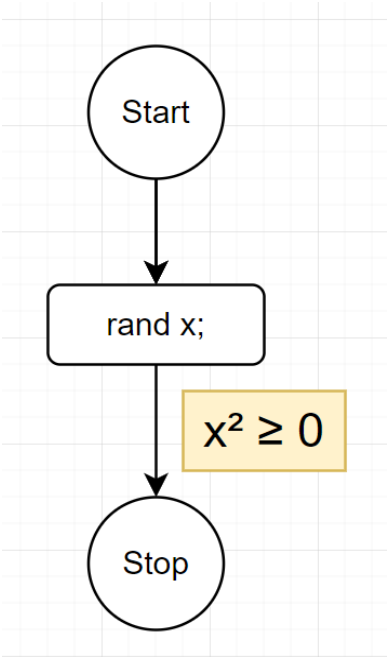






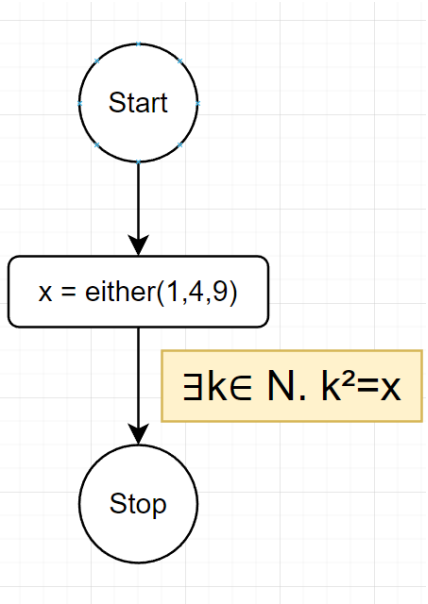
T01: MiniJava 2.0

WP[rand x;](B) =



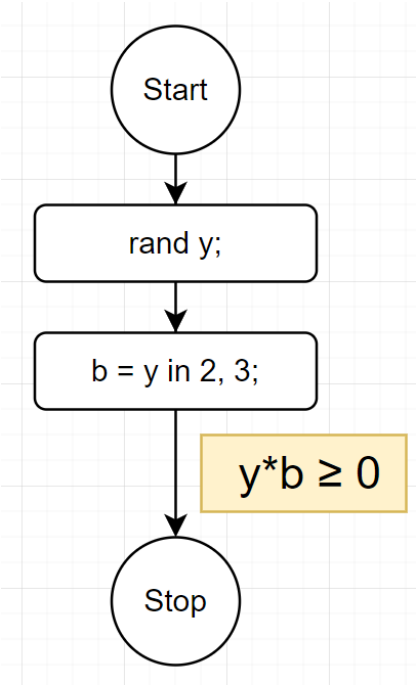
T01: MiniJava 2.0

WP[x = either e_0, e_1 … e_k](B) =



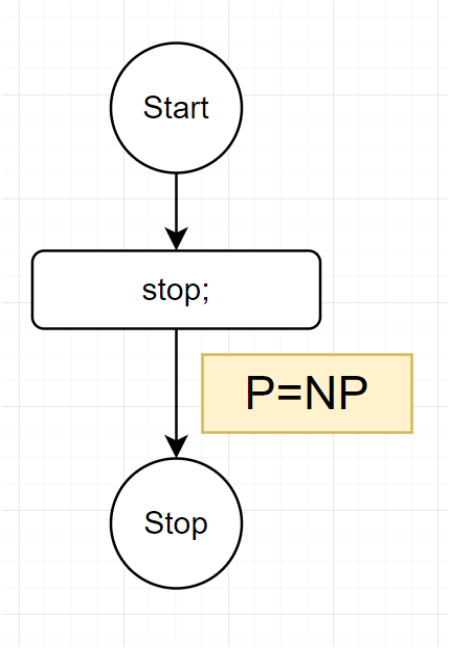
T01: MiniJava 2.0

WP[x e in a, b](B) =



T01: MiniJava 2.0

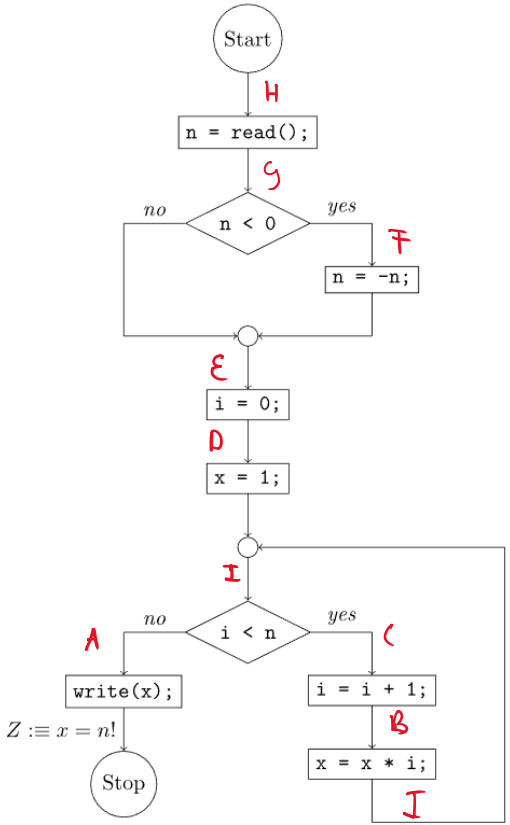
WP[stop](B) =



T02: Loop Invariants

1. Discuss the problem that arises when computing weakest preconditions to prove Z.
2. How can you use weakest preconditions to prove Z anyway?
3. Try proving Z using the the loop invariants $x \geq 0$ and $i = 0 \land x = 1 \land n = 0$ **at the end of the loop body** and in particular discuss these questions:
   - a) How has a useful loop invariant be related to Z?
   - b) What happens if the loop invariant is chosen too strong?
   - c) What happens if the loop invariant is chosen too weak?
   - d) Can you give a meaningful lower and upper bound for useful loop invariants?
4. Retry proving Z using the loop invariant $x = i!$ (again at the end of the loop body) and improve this invariant until the proof succeeds.

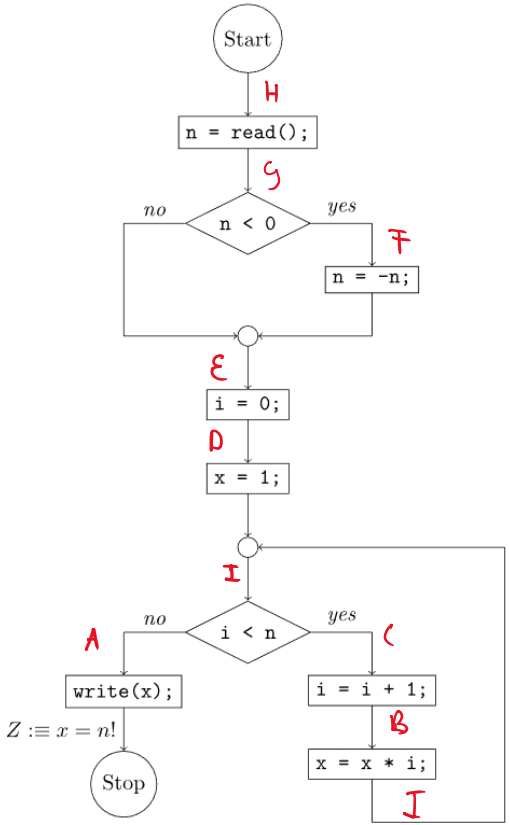
A program computes the factorial of its input:



T02: Loop Invariants 1

3. Try proving Z using the the loop invariants $x \geq 0$ and $i = 0 \wedge x = 1 \wedge n = 0$ **at the end of the loop body** and in particular discuss these questions:

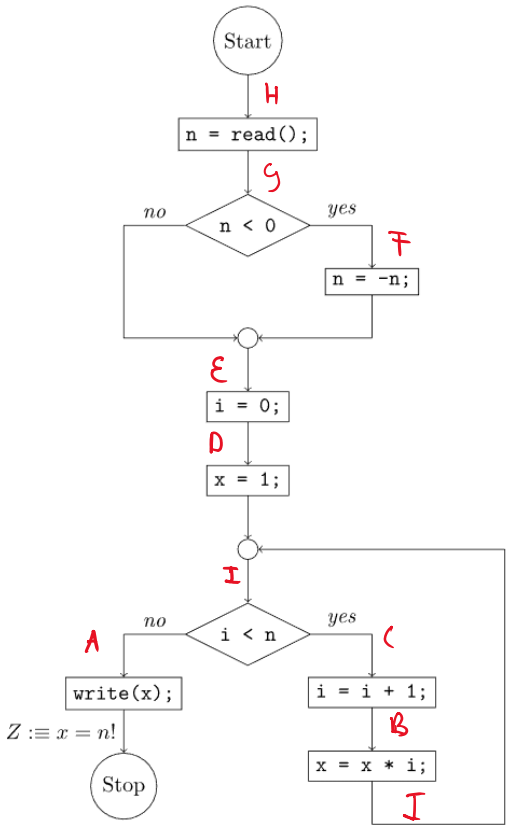
A program computes the factorial of its input:



T02: Loop Invariants 2

3. Try proving Z using the the loop invariants $x \geq 0$ and $i = 0 \wedge x = 1 \wedge n = 0$ **at the end of the loop body** and in particular discuss these questions:

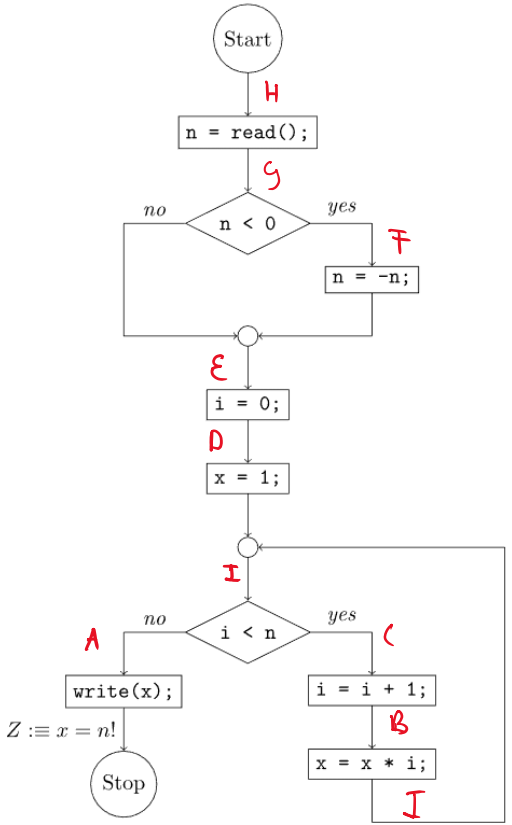
A program computes the factorial of its input:



T02: Loop Invariants 3

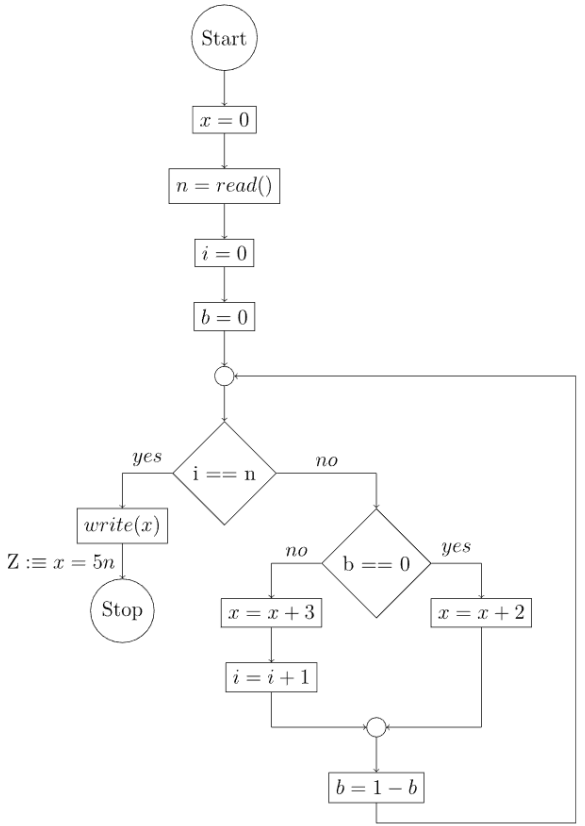
4. Retry proving Z using the loop invariant $x = i!$ (again at the end of the loop body) and improve this invariant until the proof succeeds.

A program computes the factorial of its input:



T03: Two b, or Not Two b

Prove Z using weakest preconditions.

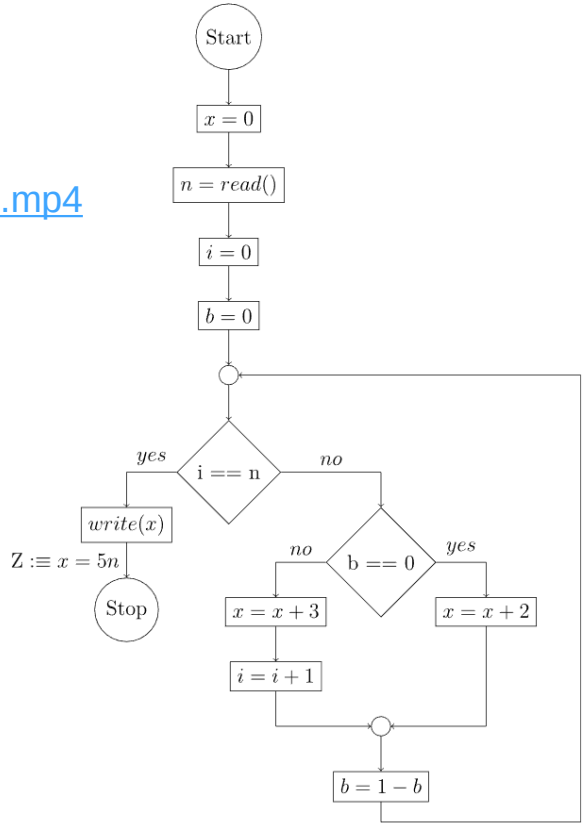


T03: Two b, or Not Two b

Tipps zum finden von Loop Invarianten:
https://ttt.in.tum.de/recordings/Info2_2017_11_24-1/Info2_2017_11_24-1.mp4

Beispieltrace: n=3

Variable \ Schleifendurchgang	0	1	2	3	4	5	6
x	**0**	**2**	**5**	**7**	**10**	**12**	**15**
i	**0**	**0**	**1**	**1**	**2**	**2**	**3**
b	**0**	**1**	**0**	**1**	**0**	**1**	**0**



Tipps für Loop Invarianten

https://ttt.in.tum.de/recordings/Info2_2017_11_24-1/Info2_2017_11_24-1.mp4

Tipp 1

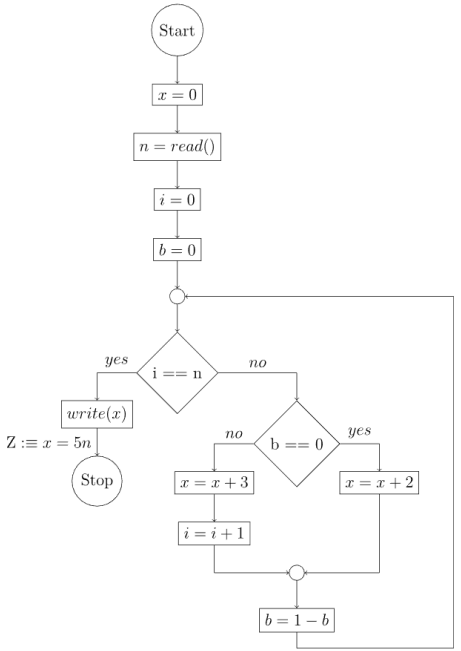
Wir benötigen eine Aussage über den Wert der Variablen, über die wir etwas beweisen wollen (x) in der Schleifeninvariante. Die Aussage muss dabei mindestens so präzise ($\neq, \geq, \leq, =$) sein, wie die Aussage, die wir beweisen wollen.

Tipp 2

Variablen, die an der Berechnung von x beteiligt sind **und** Werte von einer Schleifeniteration in die nächste transportieren ("loop-carried"), müssen in die Schleifeninvariante aufgenommen werden.

Tipp 3

Die Schleife zu verstehen ist unerlässlich. Eine Tabelle für einige Schleifendurchläufe kann helfen die Zusammenhänge der Variablen (insbesondere mit dem Schleifenzähler i) aufzudecken. Oft lassen sich mit einer Tabelle, in der man die einzelnen Berechnungsschritte notiert, diese Zusammenhänge deutlich leichter erkennen, als mit einer Tabelle, die nur konkrete Werte enthält.



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