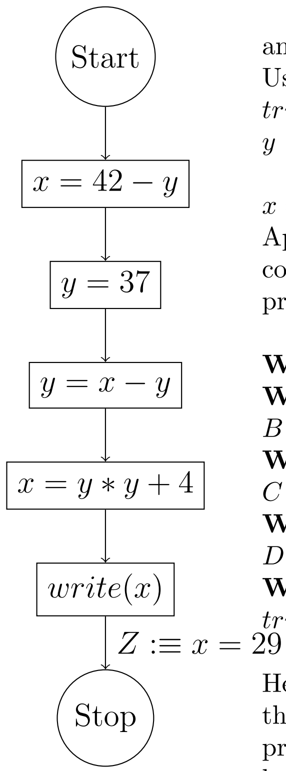
FPV Week 2 - FPV Summary for the week 2

Funktionale Programmierung (Technische Universität München)

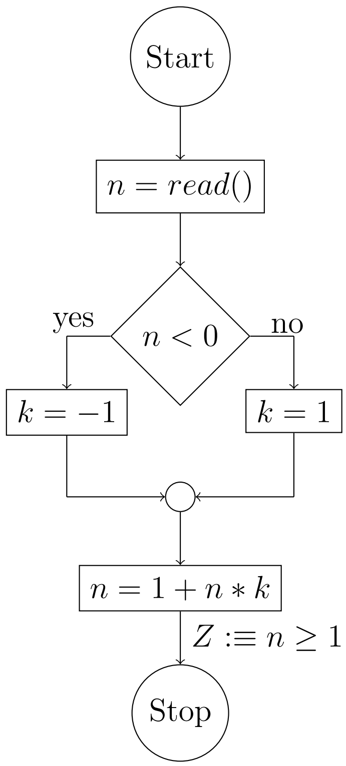


Analogously, we apply the same methods to the 2nd and the 3rd problem.
Using SP:
$true \implies x = 42 - y \implies y = 37 \implies y = x - 37 \implies x = y^2 + 4 \implies$

$x = y^2 + 4$
Apparently, we lose a lot of information using post-condition method. And we'll whether this happens with pre-condition method too.

$\mathbf{WP}[\![write(x)]\!](Z) \equiv x = 29 \equiv: A$
$\mathbf{WP}[\![x = y * y + 4]\!](A) \equiv y = 5 \lor y = -5 \equiv: B$
$\mathbf{WP}[\![y = x - y]\!](B) \equiv x - y = 5 \lor x - y = -5 \equiv: C$
$\mathbf{WP}[\![y = 37]\!](C) \equiv x = 42 \lor x = 32 \equiv: D$
$\mathbf{WP}[\![x = 42 - y]\!](D) \equiv y = 0 \lor y = 10 \not\equiv true$

Hence we can infer that the assertion Z at the end of the program does not hold, since at the beginning of the program, all assertions other than assert(true) does not hold.



Using SP:
$true \implies \forall n$
if $n < 0$: $\implies n < 0 \wedge k = -1$
else: $\implies n \geq 0 \wedge k = 1$
in decompositon: $\implies n = 1 + p$, where p is a non-negative number.
Again, using SP we can prove that Z holds. But the proof is not formal, and the process is complicated. WP can still solve the problem.
$\mathbf{WP}[\![n = 1 + n * k]\!](Z) \equiv n * k \geq 0 \equiv: A$
$\mathbf{WP}[\![k = -1]\!](A) \equiv n < 0 \equiv: B$
$\mathbf{WP}[\![k = 1]\!](A) \equiv n \geq 0 \equiv: C$
$\mathbf{WP}[\![n < 0]\!](B, C) \equiv (n < 0 \wedge n < 0) \vee (n \geq 0 \wedge n \geq 0) \equiv true \equiv: D$
$\mathbf{WP}[\![n = read()]\!](D) \equiv true$
Again, using the WP notation, we can easily prove that the final assertion holds, since the beginning assertion can be inferred as true.

3 Local Consistency

First of all, let's check out the definition of local consistency.

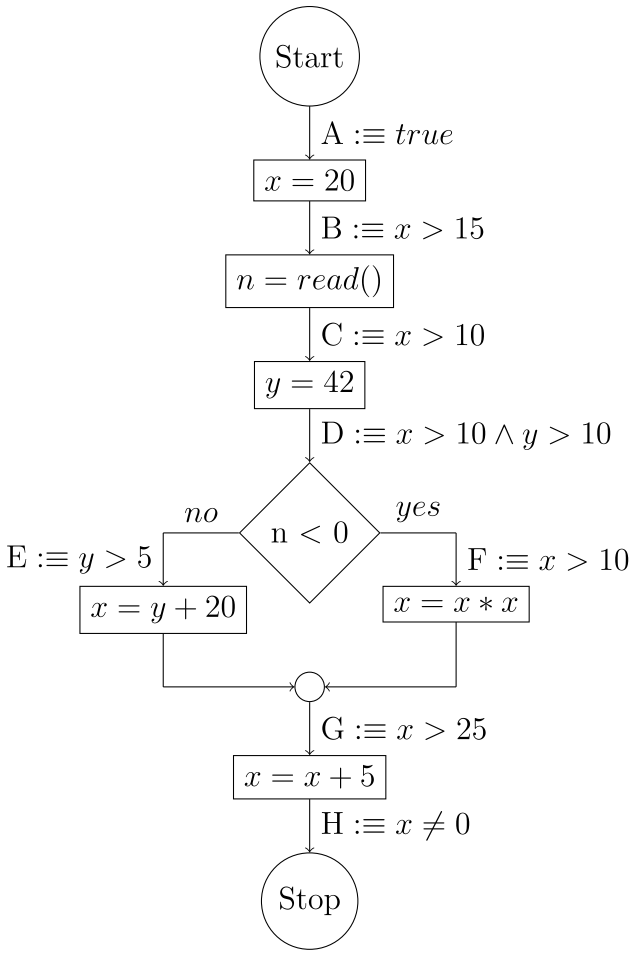
In constraint satisfaction, local consistency conditions are properties of constraint satisfaction problems related to the consistency of subsets of variables or constraints.

(Quoted from wikipedia)

To understand, it means that once we know the state of a program holds at certain point. The state of a program / assertion at the previous point, i.e. before the previous statement, can implies the WP of the current state.

To do the exercise, this means that we have to check whether the WP can be implied by the previous state.
$\mathbf{WP}[\![x = x + 5]\!](H) \equiv x + 5 \neq 0 \Longleftarrow x > 25 \equiv G$
$\mathbf{WP}[\![x = y + 20]\!](G) \equiv y + 20 > 25 \equiv y > 5 \Longleftarrow y > 5 \equiv E$
$\mathbf{WP}[\![x = x * x]\!](G) \equiv x^2 > 25 \equiv x > 5 \vee x < -5 \Longleftarrow x > 10 \equiv F$
$\mathbf{WP}[\![n < 0]\!](E, F) \equiv (n < 0 \wedge x > 10) \vee (n \leq 0 \wedge y > 5) \Longleftarrow x > 10 \wedge y > 10 \equiv D$
$\mathbf{WP}[\![y = 42]\!](D) \equiv x > 10 \wedge 42 > 10 \equiv x > 10 \Longleftarrow x > 10 \equiv C$
$\mathbf{WP}[\![n = read()]\!](C) \equiv x > 10 \Longleftarrow x > 15 \equiv B$
$\mathbf{WP}[\![x = 20]\!](B) \equiv 20 > 15 \equiv true \Longleftarrow true \equiv A$
From this perspective we can infer that the program here is local consistent even

though they don't all assert with WP at each point.



4 Trouble Sort

$\mathbf{WP}[\![b = t]\!](Z) \equiv a \leq t \leq c \equiv: A$
$\mathbf{WP}[\![a = b]\!](A) \equiv b \leq t \leq c \equiv: B$
$\mathbf{WP}[\![t = a]\!](B) \equiv b \leq a \leq c \equiv: C$
$\mathbf{WP}[\![a > b]\!](Z, C) \equiv (a \leq b \wedge a \leq b \leq c) \vee (a > b \wedge b \leq a \leq c) \equiv a \leq b \leq c \vee b \leq a \leq c \equiv a \leq c \wedge b \leq c \equiv: D$

$\mathbf{WP}[\![c = t]\!](D) \equiv a \leq t \wedge b \leq t \equiv: E$
$\mathbf{WP}[\![b = c]\!](E) \equiv a \leq t \wedge c \leq t \equiv: F$
$\mathbf{WP}[\![t = b]\!](F) \equiv a \leq b \wedge c \leq b \equiv: G$
$\mathbf{WP}[\![b > c]\!](Z, F) \equiv (a \leq b \leq c \wedge b \leq c) \vee ((a \leq b \wedge c \leq b) \wedge b > c) \equiv a \leq b \leq c \vee (a \leq b \wedge c < b) \equiv a \leq b \equiv: H$

$\mathbf{WP}[\![b = t]\!](H) \equiv a \leq t \equiv: I$
$\mathbf{WP}[\![a = b]\!](I) \equiv b \leq t \equiv: J$
$\mathbf{WP}[\![t = a]\!](J) \equiv b \leq a \equiv: K$
$\mathbf{WP}[\![a > b]\!](H, K) \equiv (a > b \wedge b \leq a) \vee (a \leq b \wedge a \leq b) \equiv true \equiv: L$
$\mathbf{WP}[\![c = read()]\!](L) \equiv true \equiv: M$
$\mathbf{WP}[\![b = read()]\!](M) \equiv true \equiv: N$
$\mathbf{WP}[\![a = read()]\!](N) \equiv true$

Now we can prove that at the beginning of the program the WP stemming from Z is true. Thus Z has to hold at the end of the program.