Vienna University of Technology WS 2010

VU Formale Methoden der Informatik

# Block 3: Formal Verification of Software

Bernhard Urban

Matr.Nr.: 0725771

lewurm@gmail.com

May 21, 2011

## Contents

# 1 Exercise 1

We want to show that the two given programs are semantically equivalent, which means that both programs deliver the same outputstate on the same inputstate $\sigma$. First, we analyze one iteration of the first program:

$$(\mathsf{while}\ e\ \mathsf{do}\ p\ \mathsf{od}, \sigma) \Rightarrow \begin{cases} (p; \mathsf{while}\ e\ \mathsf{do}\ p\ \mathsf{od}, \sigma) & \text{if } [e]\sigma \neq 0 \\ \sigma & \text{if } [e]\sigma = 0 \end{cases}$$

The execution of one iteration results into two possible states (depending on the result of $[e]\sigma$). For the second program we get:

$$(\mathsf{if}\ e\ \mathsf{then}\ p; \mathsf{while}\ e\ \mathsf{do}\ p\ \mathsf{od}\ \mathsf{else}\ \mathsf{skip}\ \mathsf{fi}, \sigma) \Rightarrow \begin{cases} (p; \mathsf{while}\ e\ \mathsf{do}\ p\ \mathsf{od}, \sigma) & \text{if } [e]\sigma \neq 0 \\ (\mathsf{skip}, \sigma) & \text{if } [e]\sigma = 0 \end{cases}$$

Obviously, both programs ends up in the same programstate after one iteration (since $(\mathsf{skip}, \sigma) \Rightarrow \sigma$). Now we know that both programs have the same behaviour on any inputstate $\sigma$ and thus they are semantically equivalent.

# 2 Exercise 2

The pre- and post-condition are obvious: $\{F : x = x_0 \wedge y = y_0\}$ and $\{G : y = x_0 \wedge x = y_0\}$

(a) Hoare calculus:

$$\frac{F \Rightarrow H'[x/x-y] \quad \{H'[x/x-y]\}x = x - y\{H'\}\ (as)}{\{F\}x = x - y\{H'\}\ (2)}\ (lc)$$

$$\frac{\{F\}x = x - y\{H'\}\ (2) \qquad \dfrac{H' \Rightarrow H[y/x+y] \quad \{H[y/x+y]\}y = x + y\{H\}\ (as)}{\{H'\}y = x + y\{H\}}\ (lc)}{\{F\}x = x - y; y = x + y\{H\}\ (1)}\ (sc)$$

$$\frac{\{F\}x = x - y; y = x + y\{H\}\ (1) \qquad \dfrac{H \Rightarrow G[x/y-x] \quad \{G[x/y-x]\}x = y - x\{G\}\ (as)}{\{H\}x = y - x\{G\}}\ (lc)}{\{F\}x = x - y; y = x + y; y = y - x\{G\}}\ (sc)$$

Now we need proper formulas for $H$ and $H'$, so that $F \Rightarrow H'[x/x-y]$, $H' \Rightarrow H[y/x+y]$ and $H \Rightarrow G[x/y-x]$ become vaild. We choose $H' \equiv H[y/x+y]$ and $H \equiv G[x/y-x]$. It remains to show $F \Rightarrow H'[x/x-y]$, i.e.:

$$\begin{aligned}
F &\Rightarrow G[x/y-x][y/x+y][x/x-y] \\
(x = x_0 \wedge y = y_0) &\Rightarrow (y = x_0 \wedge x = y_0)[x/y-x][y/x+y][x/x-y] \\
(x = x_0 \wedge y = y_0) &\Rightarrow (y = x_0 \wedge y - x = y_0)[y/x+y][x/x-y] \\
(x = x_0 \wedge y = y_0) &\Rightarrow (x + y = x_0 \wedge x + y - x = y_0)[x/x-y] \\
(x = x_0 \wedge y = y_0) &\Rightarrow (x - y + y = x_0 \wedge x - y + y - x + y = y_0) \\
(x = x_0 \wedge y = y_0) &\Rightarrow (x = x_0 \wedge y = y_0)
\end{aligned}$$

So, this implication is also valid, therefore the swap program is partially/totally correct.

(b) The annotation calculus allows us to write down the proof much cleaner:

$$
\begin{array}{l}
\{\, F\colon x = x_0 \wedge y = y_0 \,\} \\
\{\, 3\colon G[x/y-x][y/x+y][x/x-y] \,\} \\
x := x - y \\
\{\, 2\colon G[x/y-x][y/x+y] \,\} \\
y := x + y \\
\{\, 1\colon G[x/y-x] \,\} \\
x := y - x \\
\{\, G\colon y = x_0 \wedge x = y_0 \,\}
\end{array}
$$

$F \Rightarrow 3$ remains to show, which was already shown in (a).

(c) Weakest pre-condition.

$$
\begin{array}{l}
\{\, H''\colon wp(x := x - y; y := x + y; x := y - x), G \,\} \\
x := x - y \\
\{\, H'\colon wp(y := x + y; x := y - x), G \,\} \\
y := x + y \\
\{\, H\colon wp(x := y - x), G \,\} \\
x := y - x \\
\{\, G\colon y = x_0 \wedge x = y_0 \,\}
\end{array}
$$

According to the defintion $wp(v := e, F) = F[v/e]$ in the slides, $H''$ yields into $G[x/y - x][y/x + y][x/x - y]$, which is equal to our post-condition (as shown in (a)).

(d) Strongest post-condition:

$$
\begin{array}{l}
\{\, F\colon x = x_0 \wedge y = y_0 \,\} \\
x := x - y \\
\{\, H\colon sp(x := x - y, F) \,\} \\
y := x + y \\
\{\, H'\colon sp(x := x + y, H) \,\} \\
x := y - x \\
\{\, H''\colon sp(x := y - x, H') \,\}
\end{array}
$$

$H''$ evaluates to (due to $sp(v := e, F) = \exists v'(F[v/v'] \wedge v = e[v/v'])$):

$$
\begin{array}{l}
\{\, H''\colon sp(x = y - x, sp(y = x + y, sp(x = x - y, F))) \,\} \\
\Rightarrow \{\, H''\colon sp(x = y - x, sp(y = x + y, \{\exists x'(x' = x_0 \wedge y = y_0 \wedge x = x' - y\})) \,\} \\
\Rightarrow \{\, H''\colon sp(x = y - x, \{\exists y'(\exists x'(x' = x_0 \wedge y' = y_0 \wedge x = x' - y') \wedge y = x + y'\})) \,\} \\
\Rightarrow \{\, H''\colon \exists x''(\exists y'(\exists x'(x' = x_0 \wedge y' = y_0 \wedge x'' = x' - y') \wedge y = x'' + y') \wedge x = y - x'')) \,\}
\end{array}
$$

Evaluating this expression results into:

$$
\begin{array}{lll}
\{H'' : \ y = x'' + y' & \wedge\, x = y - x''\} & \\
\Rightarrow\{H'' : \ y = x' - y' + y' & \wedge\, x = y - x' - y'\} & \text{due to } x'' = x' - y' \\
\Rightarrow\{H'' : \ y = x' & \wedge\, x = y - x' - y'\} & \\
\Rightarrow\{H'' : \ y = x' & \wedge\, x = y'\} & \text{because of } y = x' \\
\Rightarrow\{H'' : \ y = x_0 & \wedge\, x = y_0\} & \text{since } x' = x_0 \text{ and } y' = y_0
\end{array}
$$

# 3 Exercise 3

In the following, these abbreviations are used:

$$
\begin{aligned}
B &:= d \cdot m \le n \\
Inv &:= 0 \le a < b \le n + 1 \wedge a \cdot m \le n < b \cdot m
\end{aligned}
$$

First, we annotate the program by applying several rules:

$\{\,1\colon m > 0 \wedge n \ge 0\,\}$
$\{\,2\colon Inv[b/n+1][a/0]\,\}$ as ↑
$a := 0;$
$\{\,3\colon Inv[b/n+1]\,\}$ as ↑
$b := n + 1;$
$\{\,4\colon Inv\,\}$ wh
while $a + 1 \ne b$ do
  $\{\,5\colon Inv \wedge a + 1 \ne b\,\}$ wh
  $\{\,6\colon (B \Rightarrow Inv[a/d]) \wedge (\neg B \Rightarrow Inv[b/d])[d/(a+b)/2]\,\}$ as ↑
  $d := (a + b)/2;$
  $\{\,7\colon (B \Rightarrow Inv[a/d]) \wedge (\neg B \Rightarrow Inv[b/d])\,\}$ if ↑
  if $d \cdot m \le n$ then
    $\{\,8\colon Inv[a/d]\,\}$ as ↑
    $a := d;$
    $\{\,9\colon Inv\,\}$ fi ↑
  else
    $\{\,10\colon Inv[b/d]\,\}$ as ↑
    $b := d;$
    $\{\,11\colon Inv\,\}$ fi ↑
  fi
  $\{\,12\colon Inv\,\}$ wh
od
$\{\,13\colon Inv \wedge a + 1 = b\,\}$ wh
$\{\,14\colon a \cdot m \le n < (a + 1) \cdot m\,\}$

Now it remains to show that $1 \Rightarrow 2$, $13 \Rightarrow 14$ and $5 \Rightarrow 6$ are indeed valid.

## $1 \Rightarrow 2$:

$$
\begin{aligned}
&m > 0 \wedge n \geq 0 \Rightarrow Inv[b/n+1][a/0] \\
&m > 0 \wedge n \geq 0 \Rightarrow 0 \leq a < b \leq n+1 \wedge a \cdot m \leq n < b \cdot m[b/n+1][a/0] \\
&m > 0 \wedge n \geq 0 \Rightarrow 0 \leq 0 < n+1 \leq n+1 \wedge 0 \cdot m \leq n < (n+1) \cdot m \\
&0 \leq n \wedge 0 < m \Rightarrow 0 < n+1 \wedge 0 \leq n < (n+1) \cdot m
\end{aligned}
$$

now we split the formula:

$$
\begin{aligned}
&n \leq 0 \Rightarrow 0 < n+1 \quad \checkmark \\
&0 < m \Rightarrow n < (n+1) \cdot m \\
&0 < m \Rightarrow n+1 \leq (n+1) \cdot m \\
&0 < m \Rightarrow 1 \leq m \quad \checkmark
\end{aligned}
$$

## $13 \Rightarrow 14$:

$$
\begin{aligned}
0 \leq a < b \leq n+1 \wedge a \cdot m \leq n < b \cdot m \wedge a+1 = b &\Rightarrow a \cdot m \leq n < (a+1) \cdot m \\
0 \leq a < a+1 \leq n+1 \wedge \underbrace{a \cdot m \leq n < (a+1) \cdot m} \wedge a+1 = b &\Rightarrow \underbrace{a \cdot m \leq n < (a+1) \cdot m} \quad \checkmark
\end{aligned}
$$

## $5 \Rightarrow 6$:

We split the proof into to parts (if and else), in order to make it more readable. This is valid, because of the relation

$$A \Rightarrow ((B \Rightarrow C) \wedge (D \Rightarrow E)) \equiv ((A \wedge B) \Rightarrow C) \wedge ((A \wedge D) \Rightarrow E)$$

### if

$$
\begin{aligned}
&0 \leq a < b \leq n+1 \wedge a \cdot m \leq n < b \cdot m \wedge a+1 \neq b \\
&\quad \Rightarrow d \cdot m \leq n \Rightarrow (0 \leq a < b \leq n+1 \wedge a \cdot m \leq n < b \cdot m)[a/d][d/(a+b)/2]
\end{aligned}
$$

substitute $a$ with $d$ and after that $d$ with $\frac{a+b}{2}$ on the right side

$$
\begin{aligned}
&0 \leq a < b \leq n+1 \wedge a \cdot m \leq n < b \cdot m \wedge a+1 \neq b \\
&\quad \Rightarrow \frac{a+b}{2} \cdot m \leq n \Rightarrow (0 \leq \frac{a+b}{2} < b \leq n+1 \wedge \frac{a+b}{2} \cdot m \leq n < b \cdot m)
\end{aligned}
$$

move $\frac{a+b}{2} \cdot m \leq n$ to the left side

$$
\begin{aligned}
&0 \leq a < b \leq n+1 \wedge a \cdot m \leq n < b \cdot m \wedge a+1 \neq b \wedge \frac{a+b}{2} \cdot m \leq n \\
&\quad \Rightarrow \underbrace{0 \leq \frac{a+b}{2} < b \leq n+1}_{c_1} \wedge \underbrace{\frac{a+b}{2} \cdot m \leq n < b \cdot m}_{c_2}
\end{aligned}
$$

At this point, we introduce

$$\begin{aligned} a &< b & a &< b \\ a+b &< b+b & a+a &< a+b \\ a+b &< 2b & 2a &< a+b \\ \frac{a+b}{2} &< b & a &< \frac{a+b}{2} \quad \text{because of } a+1 \neq b \text{ this is valid} \end{aligned}$$

$$\Rightarrow a < \frac{a+b}{2} < b \quad \ldots A_1$$

Now we show that the right side evaluates to `true` if the left side evaluates to `true` too:

- $0 \leq a < b \leq n+1 \Rightarrow c_1$ because of $A_1$ ✓
- $a \cdot m \leq n < b \cdot m \Rightarrow c_2$ because of $A_1$ ✓

**else**

$$0 \leq a < b \leq n+1 \land a \cdot m \leq n < b \cdot m \land a+1 \neq b$$
$$\Rightarrow d \cdot m > n \Rightarrow (0 \leq a < b \leq n+1 \land a \cdot m \leq n < b \cdot m)[b/d][d/(a+b)/2]$$

substitute $b$ with $d$ and after that $d$ with $\frac{a+b}{2}$ on the right side

$$0 \leq a < b \leq n+1 \land a \cdot m \leq n < b \cdot m \land a+1 \neq b$$
$$\Rightarrow \frac{a+b}{2} \cdot m > n \Rightarrow (0 \leq a < \frac{a+b}{2} \leq n+1 \land a \cdot m \leq n < \frac{a+b}{2} \cdot m)$$

move $\frac{a+b}{2} \cdot m > n$ to the left side

$$0 \leq a < b \leq n+1 \land a \cdot m \leq n < b \cdot m \land a+1 \neq b \land \frac{a+b}{2} \cdot m > n$$
$$\Rightarrow \underbrace{0 \leq a < \frac{a+b}{2} \leq n+1}_{c_1'} \land \underbrace{a \cdot m \leq n < \frac{a+b}{2} \cdot m}_{c_2'}$$

Now we show that the right side evaluates to `true` if the left side evaluates to `true` too, again using $A_1$:

- $0 \leq a < b \leq n+1 \Rightarrow c_1'$ because of $A_1$ ✓
- $a \cdot m \leq n < b \cdot m \Rightarrow c_2'$ because of $A_1$ ✓

## 3.1 Termination

Since termination is tied to the loop condition, $a+1 \neq b$ is a good starting point for the bound function $t$. Therefore, $b-a-1$ might be a suitable bound function.

while $a+1 \neq b$ do
  $\{\, 1\colon \mathit{Inv} \land a+1 \neq b \land 0 \leq t = t_0 \,\}$
  $\{\, 2\colon ((B \Rightarrow 0 \leq t < t_0[a/d]) \land (\neg B \Rightarrow 0 \leq t < t_0[b/d]))[d/\frac{a+b}{2}] \,\}$
  $d := (a+b)/2;$
  $\{\, 3\colon (B \Rightarrow 0 \leq t < t_0[a/d]) \land (\neg B \Rightarrow 0 \leq t < t_0[b/d]) \,\}$
  if $d \cdot m \leq n$ then
    $\{\, 4\colon 0 \leq t < t_0[a/d] \,\}$
    $a := d;$
    $\{\, 5\colon 0 \leq t < t_0 \,\}$
  else
    $\{\, 6\colon 0 \leq t < t_0[b/d] \,\}$
    $b := d;$
    $\{\, 7\colon 0 \leq t < t_0 \,\}$
  fi
  $\{\, 8\colon 0 \leq t < t_0 \,\}$
od

It remains to show that $1 \Rightarrow 2$ is valid. Again, we split the proof into two parts (if and else).

### $1 \Rightarrow 2$:

**if**

$$0 \leq a < b \leq n+1 \land a \cdot m \leq n < b \cdot m \land a+1 \neq b \land 0 \leq t_0$$
$$\Rightarrow (d \cdot m \leq n \Rightarrow (0 \leq b-a-1 < t_0)[a/d])[d/\frac{a+b}{2}]$$
$$0 \leq a < b \leq n+1 \land a \cdot m \leq n < b \cdot m \land a+1 \neq b \land 0 \leq t_0$$
$$\Rightarrow \frac{a+b}{2} \cdot m \leq n \Rightarrow 0 \leq b - \frac{a+b}{2} - 1 < t_0$$
$$0 \leq a < b \leq n+1 \land a \cdot m \leq n < b \cdot m \land a+1 \neq b \land 0 \leq t_0 \land \frac{a+b}{2} \cdot m \leq n$$
$$\Rightarrow 0 \leq b - \frac{a+b}{2} - 1 < t_0$$
$$0 \leq a < b \leq n+1 \land a \cdot m \leq n < b \cdot m \land a+1 \neq b \land 0 \leq t_0 \land \frac{a+b}{2} \cdot m \leq n$$
$$\Rightarrow 0 \leq \frac{2b-a-b}{2} - 1 < t_0 \quad \checkmark \quad \text{since } a \geq 0 \Rightarrow t \text{ decreases}$$

**else**

$$
\begin{aligned}
0 \leq& a < b \leq n+1 \land a \cdot m \leq n < b \cdot m \land a+1 \neq b \land 0 \leq t_0 \\
&\Rightarrow (d \cdot m > n \Rightarrow (0 \leq b-a-1 < t_0)[b/d])[d/\frac{a+b}{2}] \\
0 \leq& a < b \leq n+1 \land a \cdot m \leq n < b \cdot m \land a+1 \neq b \land 0 \leq t_0 \\
&\Rightarrow \frac{a+b}{2} \cdot m > n \Rightarrow 0 \leq \frac{a+b}{2} - a - 1 < t_0 \\
0 \leq& a < b \leq n+1 \land a \cdot m \leq n < b \cdot m \land a+1 \neq b \land 0 \leq t_0 \land \frac{a+b}{2} \cdot m > n \\
&\Rightarrow 0 \leq \frac{a+b}{2} - a - 1 < t_0 \\
0 \leq& a < b \leq n+1 \land a \cdot m \leq n < b \cdot m \land a+1 \neq b \land 0 \leq t_0 \land \frac{a+b}{2} \cdot m > n \\
&\Rightarrow 0 \leq \frac{a+b-2a}{2} - 1 < t_0 \quad \checkmark \quad \text{since } a \geq 0 \Rightarrow t \text{ decreases}
\end{aligned}
$$

# 4 Exercise 4

By looking at sp(if $e$ then $p$ else $q$ fi, $F$) and the if$\downarrow$ rule

$$\{F\} \text{ if } e \text{ then } \ldots \text{ else} \mapsto \{F\} \text{ if } e \text{ then } \{F \land e\} \ \ldots \text{ else } \{F \land \neg e\} \quad \text{if}\downarrow$$

plus some intuition, we get

- if: $\{e \land F\}p\{\text{sp}(p, \{e \land F\}\}$
- else: $\{\neg e \land F\}p\{\text{sp}(q, \{\neg e \land F\}\}$

as strongest post-condition for each branch. By applying the dual-fi rule

$$\{F\} \text{ else } \ldots \ \{G\} \mapsto \{F\} \text{ else } \ldots \ \{G\} \text{ fi } \{F \lor G\} \quad \text{fi}\downarrow$$

we can conclude:

$$\text{sp}(\text{if } e \text{ then } p \text{ else } q \text{ fi}, F) = \{\text{sp}(p, \{e \land F\}) \lor \text{sp}(q, \{\neg e \land F\})\}$$

# 5 Exercise 5

## $\{F\}$ $p$ $\{$**true**$\}$

(a) $\{ F\colon x \neq 0 \}$ so $p$ may not terminate, depending on whether $x$ is even or odd.

(b) $\{ F\colon x = 0 \} \Rightarrow$ the program always terminate.

### $\{F\}\ p\ \{\textsf{false}\}$

(a) $\{\, F\colon x = 1\,\}$ so the program doesn't terminate obviously.

(b) No formula, since the post-condition is false, which means the program never terminate.

### $\{\textsf{true}\}\ p\ \{F\}$

(a) $\{\, F\colon x = 0\,\}$ when the program terminates, the post-condition is fulfilled.

(b) No formula, because we don't know anything about $x$, therefore we can end up an infinte loop.

### $\{\textsf{false}\}\ p\ \{F\}$

(a) $\{\, F\colon x = 0\,\}$ even when $p$ would terminate, it would fullfil the post-condition then.

(b) No formula, since the pre-condition is false anyway and therefore termination won't be guaranteed.

# 6 Exercise 6

An assertion $\{F\}p\{G\}$ is totally correct, if

- whenever $p$ starts in an $F$-State, then $p$ terminates and stops in a $G$-State.
- $\forall \sigma \in \mathcal{S} : [F]\sigma \Rightarrow \mathrm{def}([p]\sigma) \wedge [G][p]\sigma$

Looking at the premise $\{F \wedge e\}p\{G\}$, which must be totally correct, we can rewrite it according to the definition above as (we use $\boxed{\sigma : F}$ as an abbreviation for "$\sigma$ is a defined $F$-State", i.e., for "$\sigma$ is a defined state and the formula $F$ is true in $\sigma$."):

$$\forall \sigma \in \mathcal{S} : [F \wedge e]\sigma \Rightarrow \mathrm{def}([p]\sigma) \wedge [G][p]\sigma$$
$$\forall \sigma \in \mathcal{S} : [F]\sigma \wedge [e]\sigma \Rightarrow \mathrm{def}([p]\sigma) \wedge [G][p]\sigma$$
$$\forall \sigma \in \mathcal{S} : [F]\sigma \wedge [e]\sigma \Rightarrow \boxed{[p]\sigma : G}$$
$$\forall \sigma \in \mathcal{S} : [F]\sigma \wedge [e]\sigma \neq 0 \Rightarrow \boxed{[p]\sigma : G} \qquad \text{since } [e]\sigma \text{ is a boolean expression}$$
$$\forall \sigma \in \mathcal{S} : [F]\sigma \Rightarrow [e]\sigma \neq 0 \Rightarrow \boxed{[p]\sigma : G} \qquad \text{moving it to the right side}$$

Similar, $\{F \wedge \neg e\}q\{G\}$ results into

$$\forall \sigma \in \mathcal{S} : [F]\sigma \Rightarrow [e]\sigma = 0 \Rightarrow \boxed{[q]\sigma : G}$$

Now we take a look at the if statement for TPL (natural semantics):

$$[\textsf{if } e \textsf{ then } p \textsf{ else } q \textsf{ fi}]\sigma \Rightarrow \begin{cases} [p]\sigma & \text{if } [e]\sigma \neq 0 \\ [q]\sigma & \text{if } [e]\sigma = 0 \end{cases}$$

This rule allows us to rewrite the both statements above such as:

$$\forall \sigma \in \mathcal{S} : [F]\sigma \Rightarrow \boxed{[\textsf{if } e \textsf{ then } p \textsf{ else } q \textsf{ fi}]\sigma : G}$$
$$\forall \sigma \in \mathcal{S} : [F]\sigma \Rightarrow \text{def}([\textsf{if } e \textsf{ then } p \textsf{ else } q \textsf{ fi}]\sigma) \wedge [G][\textsf{if } e \textsf{ then } p \textsf{ else } q \textsf{ fi}]\sigma$$
$$\Rightarrow \{F\}\textsf{if } e \textsf{ then } p \textsf{ else } q \textsf{ fi}\{G\} \quad \checkmark \quad \text{by using the defintion of totally correctness}$$

# 7 Exercise 7

By looking at the post-condition $\{0 \leq y^2 \leq x < (y+1)^2\}$, we can easily identify an invariant for our program, since we know that $y$ must be at least 0 (otherwise we would get complex numbers): $\{0 \leq y^2 \leq x\}$. Due to the wh-rule

$$\textsf{while } e \textsf{ do } \ldots \textsf{ od } \mapsto \{Inv\} \textsf{ while } e \textsf{ do}\{Inv \wedge e\} \ldots \{Inv\} \textsf{ od } \{Inv \wedge \neg e\} \quad \textsf{wh}$$

we know, that the post-condition of a while consists of the conjunction of the invariant and the negated loop-condition. Therefore

$$\neg e = x < (y+1)^2$$
$$e = x \geq (y+1)^2$$

is the loop-condition. In order to guarantee termination, we consider the loop condition. We see, that $y$ have to increase over time. Also this is the closest condition which we can get, since we're operating on integer numbers.
Thus, a simple algorithm would be (although it isn't quite efficient):

$\{\, 2\colon x \geq 0 \,\}$
$y := 0;$
$\{\, Inv\colon 0 \leq y^2 \leq x \,\}$
while $x \geq (y+1)^2$ do
  $y := y + 1;$
od
$\{\, 1\colon 0 \leq y^2 \leq x < (y+1)^2 \,\}$