StuDocu.com

Folien-2 fpv in technische universitat munchen

Funktionale Programmierung (Technische Universität München)

StuDocu wird von keiner Universität gesponsert oder unterstützt.
Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Example

assignment:	`x = x-y;`
post-condition:	$x > 0$
weakest pre-condition:	$x - y > 0$
stronger pre-condition:	$x - y > 2$
even stronger pre-condition:	$x - y = 3$

This document is available free of charge on StuDocu.com
Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

… in the GCD Program (1):

assignment: `x = x-y;`

post-condition: A

weakest pre-condition:

$$
\begin{aligned}
A[x-y/x] &\equiv gcd(a,b) = gcd(x-y,y) \\
&\equiv gcd(a,b) = gcd(x,y) \\
&\equiv A
\end{aligned}
$$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

... in the GCD Program (2):

assignment: `y = y-x;`

post-condition: A

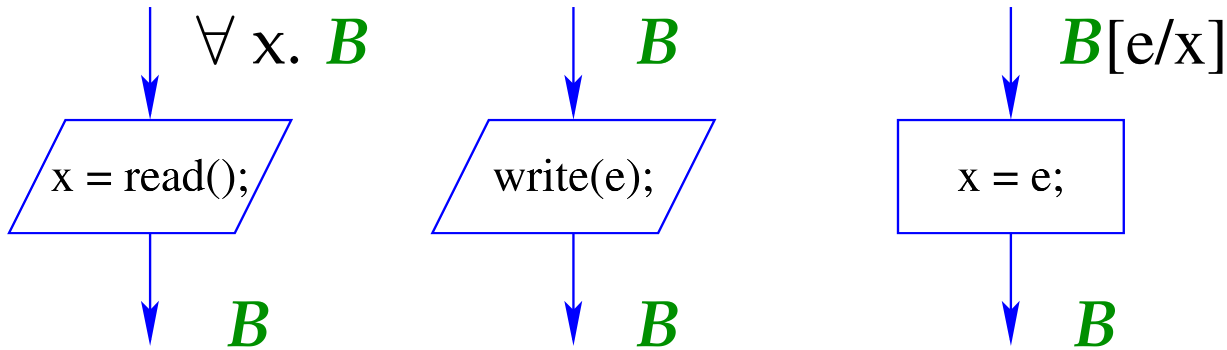
weakest pre-condition:

$$
\begin{array}{lcl}
A[y-x/y] & \equiv & gcd(a,b) = gcd(x, y-x) \\
 & \equiv & gcd(a,b) = gcd(x,y) \\
 & \equiv & A
\end{array}
$$

This document is available free of charge on StuDocu.com

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Wrap-up



$$
\begin{aligned}
\mathbf{WP}[\![\texttt{;}]\!](B) &\equiv B \\
\mathbf{WP}[\![\texttt{x = e;}]\!](B) &\equiv B[e/x] \\
\mathbf{WP}[\![\texttt{x = read();}]\!](B) &\equiv \forall\, x.\, B \\
\mathbf{WP}[\![\texttt{write(e);}]\!](B) &\equiv B
\end{aligned}
$$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

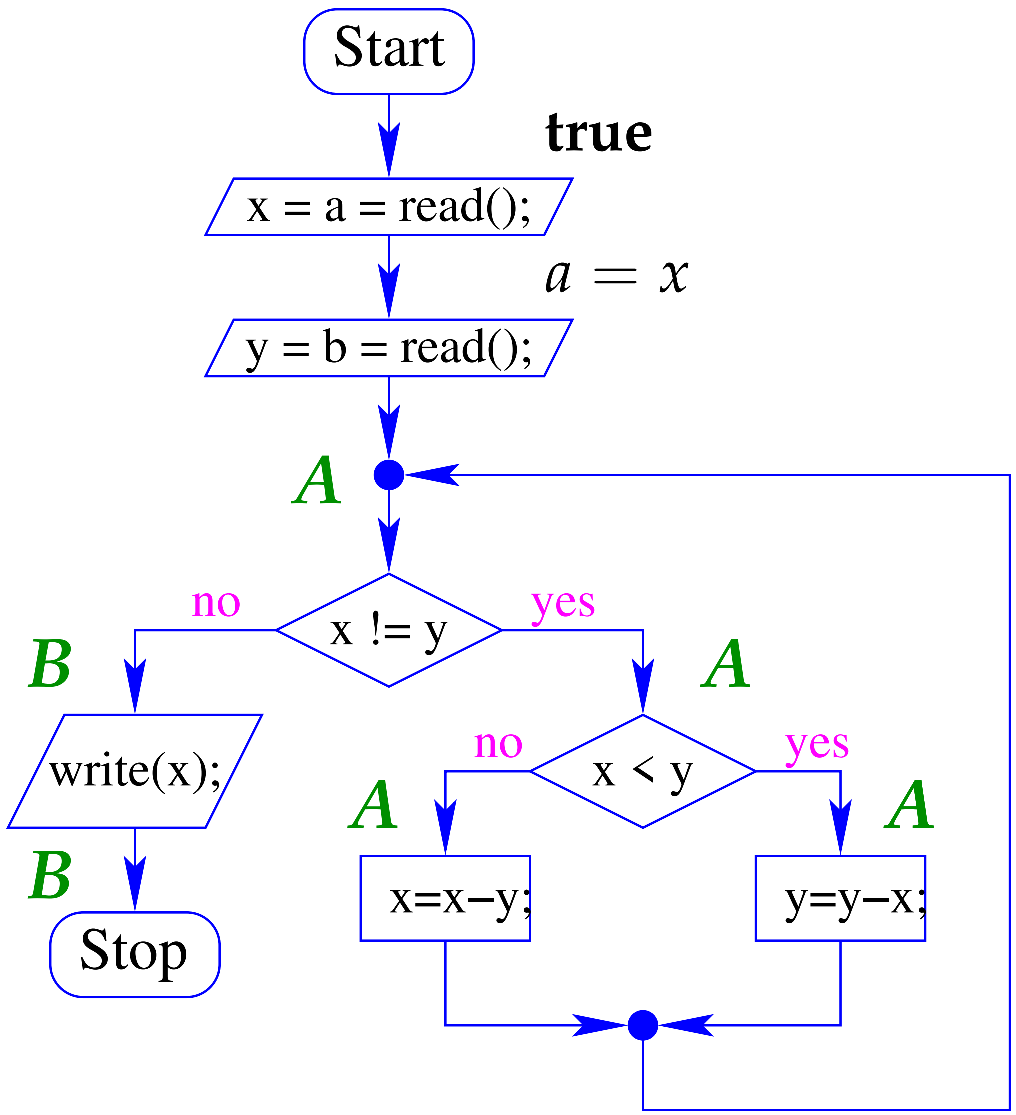
Discussion

- For all actions, the wrap-up provides the corresponding weakest pre-conditions for a post-condition B.
- An output statement does not change any variable. Therefore, the weakest pre-condition is B itself.
- An input statement `x=read();` modifies the variable `x` unpredictably.

  In order B to hold after the input, B must hold for every possible `x` before the input.

This document is available free of charge on StuDocu.com
Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Orientation



Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

For the statements: `b = read(); y = b;` we calculate:

$$\begin{aligned} \mathbf{WP}[\![\texttt{y = b;}]\!]\,(A) &\equiv A[b/y] \\ &\equiv gcd(a,b) = gcd(x,b) \end{aligned}$$

This document is available free of charge on StuDocu.com
Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

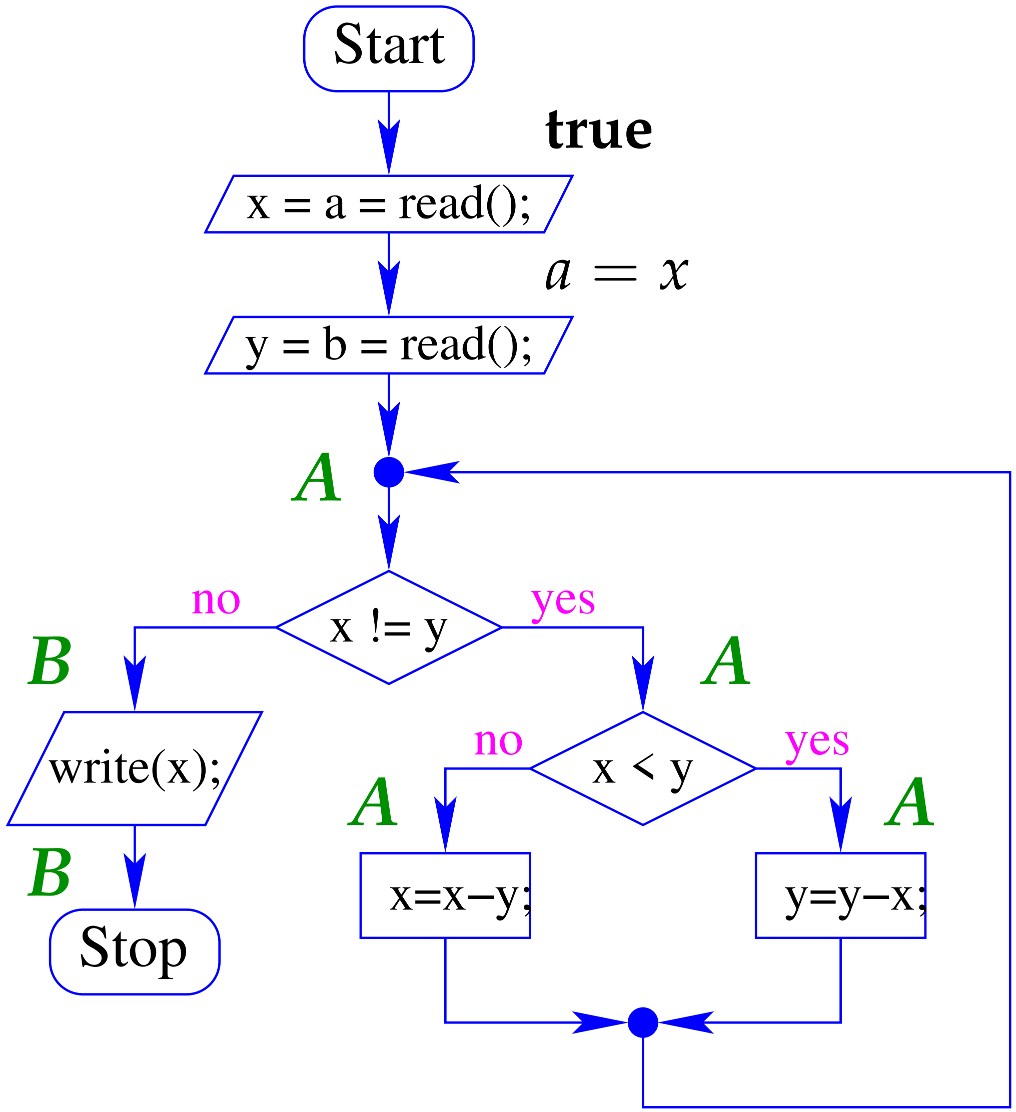
For the statements: `b = read(); y = b;` we calculate:

$$\begin{array}{lll}\mathbf{WP}[\![\texttt{y = b;}]\!]\,(A) & \equiv & A[b/y] \\ & \equiv & gcd(a,b) = gcd(x,b)\end{array}$$

$$\begin{array}{lll}\mathbf{WP}[\![\texttt{b = read();}]\!]\,(gcd(a,b) = gcd(x,b)) & & \\ & \equiv & \forall\, b.\; gcd(a,b) = gcd(x,b) \\ & \Leftarrow & a = x\end{array}$$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Orientation



This document is available free of charge on StuDocu.com
Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

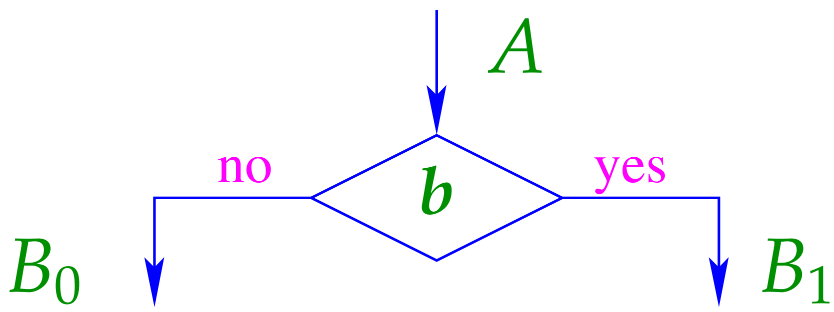
For the statements: `a = read(); x = a;` we calculate:

$$\begin{aligned}\mathbf{WP}[\![\texttt{x = a;}]\!]\,(a = x) &\equiv a = a \\ &\equiv \mathbf{true}\end{aligned}$$

$$\begin{aligned}\mathbf{WP}[\![\texttt{a = read();}]\!]\,(\mathbf{true}) &\equiv \forall\, a.\ \mathbf{true} \\ &\equiv \mathbf{true}\end{aligned}$$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Sub-problem 2: Conditionals



It should hold:

- $A \wedge \neg b \Rightarrow B_0$ and
- $A \wedge b \Rightarrow B_1$.

This document is available free of charge on StuDocu.com
Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

This is the case, if A implies the weakest pre-condition of the conditional branching:

$$\mathbf{WP}[\![b]\!]\,(B_0, B_1) \;\equiv\; ((\neg b) \Rightarrow B_0) \;\wedge\; (b \Rightarrow B_1)$$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

This is the case, if A implies the weakest pre-condition of the conditional branching:

$$\mathbf{WP}[\![b]\!]\,(B_0, B_1) \;\equiv\; ((\neg b) \Rightarrow B_0) \;\wedge\; (b \Rightarrow B_1)$$

The weakest pre-condition can be rewritten into:

$$\begin{aligned}
\mathbf{WP}[\![b]\!]\,(B_0, B_1) \quad &\equiv \quad (b \vee B_0) \wedge (\neg b \vee B_1) \\
&\equiv \quad (\neg b \wedge B_0) \vee (b \wedge B_1) \vee (B_0 \wedge B_1) \\
&\equiv \quad (\neg b \wedge B_0) \vee (b \wedge B_1)
\end{aligned}$$

This document is available free of charge on StuDocu.com

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Example

$$B_0 \equiv x > y \land y > 0 \qquad\qquad B_1 \equiv y > x \land x > 0$$

Assume that b is the condition $y > x$.

Then the weakest pre-condition is given by:

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Example

$$B_0 \equiv x > y \wedge y > 0 \qquad\qquad B_1 \equiv y > x \wedge x > 0$$

Assume that b is the condition $y > x$.

Then the weakest pre-condition is given by:

$$\begin{aligned}
(x \geq y \wedge x > y \wedge y > 0) \vee (y > x \wedge y > x \wedge x > 0) \\
\equiv \quad (x > y \wedge y > 0) \vee (y > x \wedge x > 0) \\
\equiv \quad x > 0 \wedge y > 0 \wedge x \neq y
\end{aligned}$$

This document is available free of charge on StuDocu.com

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

… for the GCD Example

$$
\begin{aligned}
b &\equiv y > x \\
\neg b \wedge A &\equiv x \geq y \wedge gcd(a,b) = gcd(x,y) \\
b \wedge A &\equiv y > x \wedge gcd(a,b) = gcd(x,y)
\end{aligned}
$$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

... for the GCD Example

$$
\begin{aligned}
b &\equiv y > x \\
\neg b \wedge A &\equiv x \geq y \wedge gcd(a,b) = gcd(x,y) \\
b \wedge A &\equiv y > x \wedge gcd(a,b) = gcd(x,y)
\end{aligned}
$$

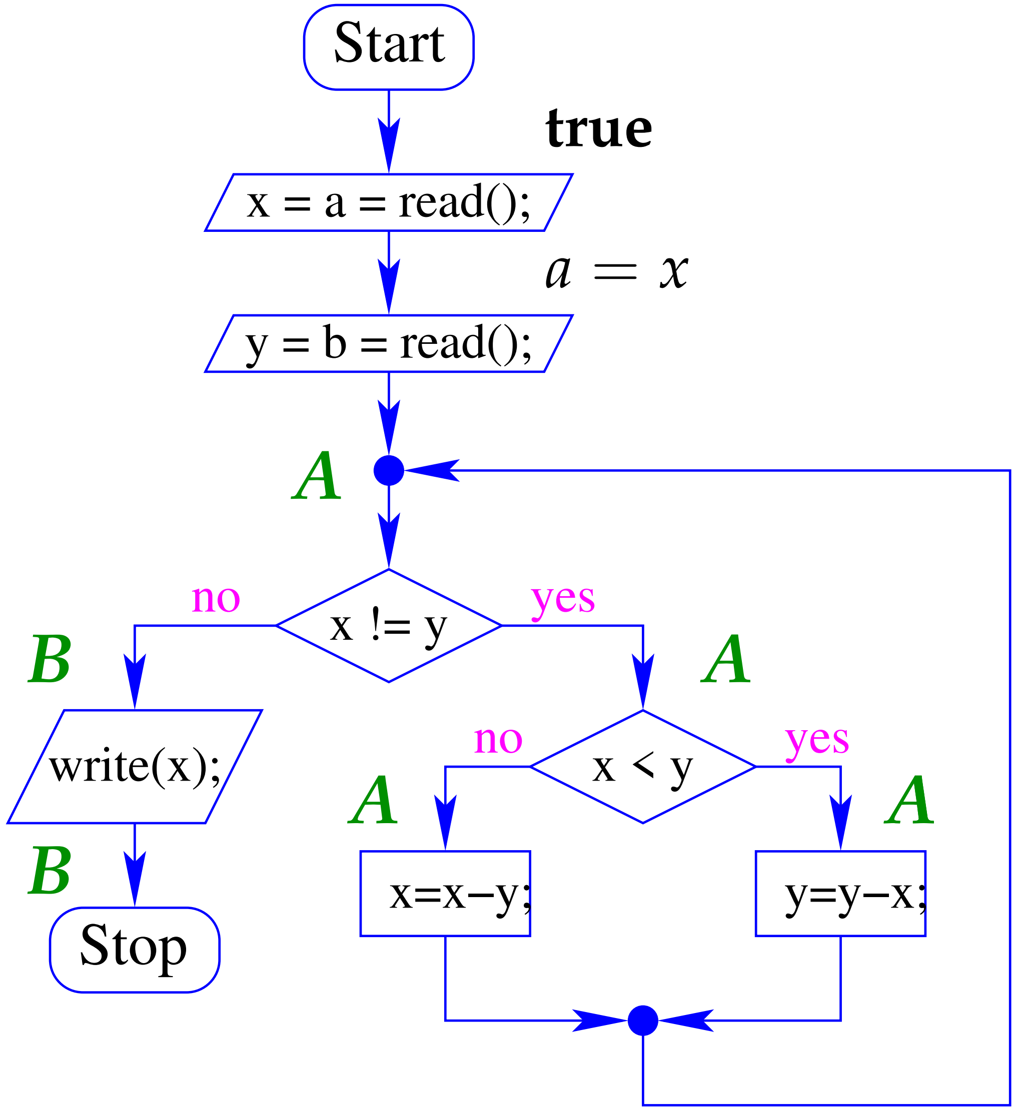
$\Longrightarrow$ The weakest pre-condition is given by

$$gcd(a,b) = gcd(x,y)$$

... i.e., exactly A

This document is available free of charge on StuDocu.com
Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Orientation



Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

The argument for the assertion before the loop is analogous:

$$\begin{aligned} b &\equiv y \neq x \\[1em] \neg b \wedge B &\equiv B \\ b \wedge A &\equiv A \wedge x \neq y \end{aligned}$$

$\Longrightarrow$ $A \equiv (A \wedge x = y) \vee (A \wedge x \neq y)$ is the weakest pre-condition for the conditional branching.

This document is available free of charge on StuDocu.com
Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Summary of the Approach

- Annotate each program point with an assertion.
- Program start should receive annotation **true**.
- Verify for each statement s between two assertions A and B, that A implies the weakest pre-condition of s for B i.e.,

$$A \Rightarrow \mathbf{WP}[\![s]\!](B)$$

- Verify for each conditional branching with condition b, whether the assertion A before the condition implies the weakest pre-condition for the post-conditions B_0 and B_1 of the branching, i.e.,

$$A \Rightarrow \mathbf{WP}[\![b]\!]\ (B_0, B_1)$$

An annotation with the last two properties is called locally consistent.

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

1.2 Correctness

Questions

- Which program properties can be verified by means of locally consistent annotations ?
- How can we be sure that our method does not prove wrong claims ??

This document is available free of charge on StuDocu.com

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Recap (1)

- In MiniJava, the program state σ consists of a variable assignment, i.e., a mapping of program variables to integers (their values), e.g.,

$$\sigma = \{x \mapsto 5, y \mapsto -42\}$$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Recap (1)

- In MiniJava, the program state σ consists of a variable assignment, i.e., a mapping of program variables to integers (their values), e.g.,

$$\sigma = \{x \mapsto 5, y \mapsto -42\}$$

- A state σ satisfies an assertion A , if

$$A[\sigma(x)/x]_{x \in A}$$

$//$ every variable in A is substituted by its value in σ

is a tautology, i.e., equivalent to **true**.

We write: $\sigma \models A$.

This document is available free of charge on StuDocu.com
Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Example

$$
\begin{aligned}
\sigma &= \{x \mapsto 5, y \mapsto 2\} \\
A &\equiv (x > y) \\
A[5/x, 2/y] &\equiv (5 > 2) \\
&\equiv \mathbf{true}
\end{aligned}
$$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Example

$$
\begin{aligned}
\sigma &= \{x \mapsto 5, y \mapsto 2\} \\
A &\equiv (x > y) \\
A[5/x, 2/y] &\equiv (5 > 2) \\
&\equiv \textbf{true}
\end{aligned}
$$

$$
\begin{aligned}
\sigma &= \{x \mapsto 5, y \mapsto 12\} \\
A &\equiv (x > y) \\
A[5/x, 12/y] &\equiv (5 > 12) \\
&\equiv \textbf{false}
\end{aligned}
$$

This document is available free of charge on StuDocu.com

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Trivial Properties

$$\sigma \models \mathbf{true} \quad \text{for every} \quad \sigma$$
$$\sigma \models \mathbf{false} \quad \text{for no} \quad \sigma$$

$\sigma \models A_1$ and $\sigma \models A_2$ is equivalent to
$$\sigma \models A_1 \wedge A_2$$

$\sigma \models A_1$ or $\sigma \models A_2$ is equivalent to
$$\sigma \models A_1 \vee A_2$$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Recap (2)

- An execution trace π traverses a path in the control-flow graph.
- It starts in a program point u_0 with an initial state σ_0 and leads to a program point u_m with a final state σ_m.
- Every step of the execution trace performs an action and (possibly) changes program point and state.

This document is available free of charge on StuDocu.com

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Recap (2)

- An execution trace π traverses a path in the control-flow graph.
- It starts in a program point u_0 with an initial state σ_0 and leads to a program point u_m with a final state σ_m.
- Every step of the execution trace performs an action and (possibly) changes program point and state.

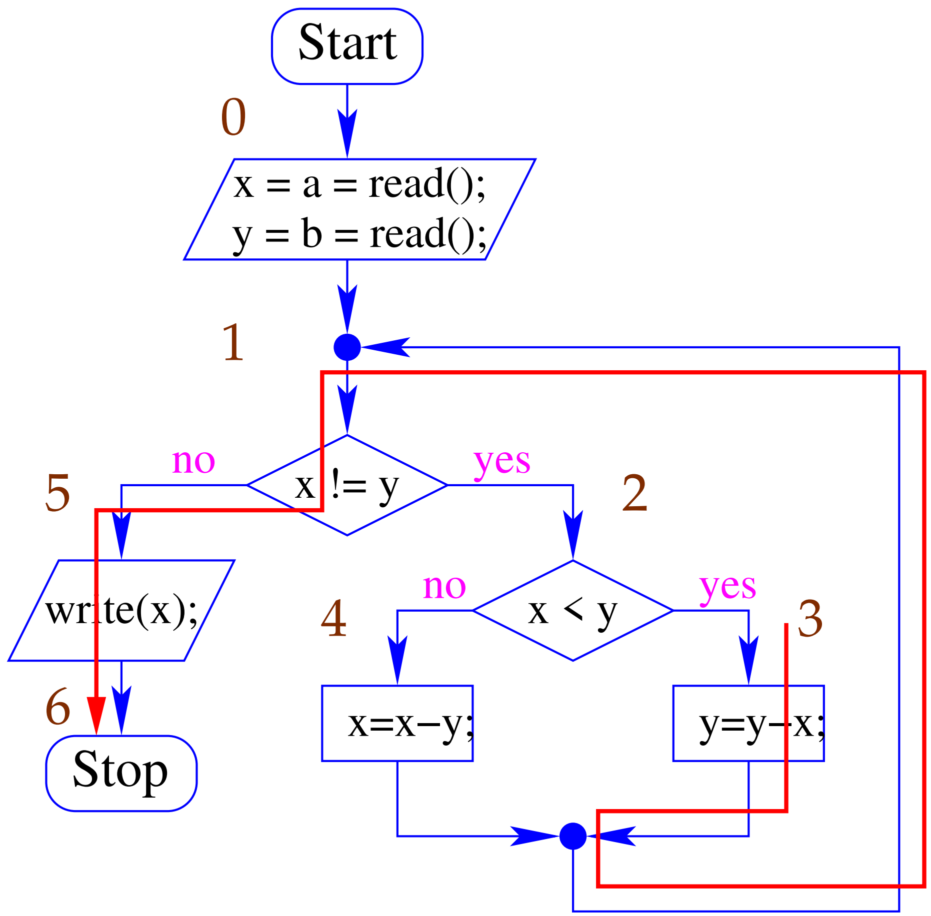
  $\Longrightarrow$ The trace π can be represented as a sequence

$$(u_0, \sigma_0) s_1 (u_1, \sigma_1) \ldots s_m (u_m, \sigma_m)$$

  where s_i are elements of the control-flow graph, i.e., basic statements or (possibly negated) conditional expressions (guards) ...

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Example



This document is available free of charge on StuDocu.com
Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Assume that we start in point 3 with $\{x \mapsto 6, y \mapsto 12\}$.

Then we obtain the following execution trace:

$$\begin{array}{lll} \pi & = & (3, \{x \mapsto 6, y \mapsto 12\}) \quad \texttt{y = y-x;} \\ & & (1, \{x \mapsto 6, y \mapsto 6\}) \quad \neg\texttt{(x != y)} \\ & & (5, \{x \mapsto 6, y \mapsto 6\}) \quad \texttt{write(x);} \\ & & (6, \{x \mapsto 6, y \mapsto 6\}) \end{array}$$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Assume that we start in point 3 with $\{x \mapsto 6, y \mapsto 12\}$.

Then we obtain the following execution trace:

$$\begin{array}{lcll} \pi & = & (3, \{x \mapsto 6, y \mapsto 12\}) & \texttt{y = y-x;} \\ & & (1, \{x \mapsto 6, y \mapsto 6\}) & \neg\texttt{(x != y)} \\ & & (5, \{x \mapsto 6, y \mapsto 6\}) & \texttt{write(x);} \\ & & (6, \{x \mapsto 6, y \mapsto 6\}) & \end{array}$$

Important operation: Update of of state

$$\sigma \oplus \{x \mapsto d\} = \{z \mapsto \sigma\, z \mid z \not\equiv x\} \cup \{x \mapsto d\}$$

This document is available free of charge on StuDocu.com

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Assume that we start in point 3 with $\{x \mapsto 6, y \mapsto 12\}$.

Then we obtain the following execution trace:

$$
\begin{array}{rcll}
\pi & = & (3, \{x \mapsto 6, y \mapsto 12\}) & \texttt{y = y-x;} \\
 & & (1, \{x \mapsto 6, y \mapsto 6\}) & \neg\texttt{(x != y)} \\
 & & (5, \{x \mapsto 6, y \mapsto 6\}) & \texttt{write(x);} \\
 & & (6, \{x \mapsto 6, y \mapsto 6\}) &
\end{array}
$$

Important operation: Update of of state

$$\sigma \oplus \{x \mapsto d\} \;=\; \{z \mapsto \sigma\, z \mid z \not\equiv x\} \cup \{x \mapsto d\}$$

$$\{x \mapsto 6, y \mapsto 12\} \oplus \{y \mapsto 6\} \;=\; \{x \mapsto 6, y \mapsto 6\}$$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Theorem

Let p be a MiniJava program, let π be an execution trace starting in program point u and leading to program point v.

Assumptions:

- The program points in p are annotated by assertions which are locally consistent.
- The program point u is annotated with A.
- The program point v is annotated with B.

This document is available free of charge on StuDocu.com

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Theorem

Let p be a MiniJava program, let π be an execution trace starting in program point u and leading to program point v.

Assumptions:

- The program points in p are annotated by assertions which are locally consistent.
- The program point u is annotated with A.
- The program point v is annotated with B.

Conclusion:

If the initial state of π satisfies the assertion A, then the final state satisfies the assertion B.

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Remarks

- If the start point of the program is annotated with **true**, then every execution trace reaching program point v satisfies the assertion at v.
- In order to prove that an assertion A holds at a program point v, we require a locally consistent annotation satisfying:

  (1) The start point is annotated with **true**.

  (2) The assertion at v implies A.

This document is available free of charge on StuDocu.com
Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Remarks

- If the start point of the program is annotated with **true**, then every execution trace reaching program point v satisfies the assertion at v.
- In order to prove that an assertion A holds at a program point v, we require a locally consistent annotation satisfying:

  (1) The start point is annotated with **true**.

  (2) The assertion at v implies A.
- So far, our method does not provide any guarantee that v is ever reached !!!
- If a program point v can be annotated with the assertion **false**, then v cannot be reached.

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Proof

Let $\pi = (u_0, \sigma_0) s_1 (u_1, \sigma_1) \ldots s_m (u_m, \sigma_m)$

Assumption: $\sigma_0 \models A$.

Proof obligation: $\sigma_m \models B$.

Idea

Induction on the length m of the execution trace.

This document is available free of charge on StuDocu.com

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Proof

Let $\pi = (u_0, \sigma_0) s_1 (u_1, \sigma_1) \ldots s_m (u_m, \sigma_m)$

Assumption: $\sigma_0 \models A$.

Proof obligation: $\sigma_m \models B$.

Idea

Induction on the length m of the execution trace.

Base $\boxed{m = 0\textbf{:}}$

The endpoint of the execution equals the startpoint.

$\Longrightarrow$ $\sigma_0 = \sigma_m$ and $A \equiv B$

$\Longrightarrow$ the claim holds.

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Important Notion: Evaluation of Expressions

Program State

$$\sigma \quad = \quad \{x \mapsto 5, y \mapsto -1, z \mapsto 21\}$$

Arithmetic Expression

$$t \quad \equiv \quad 2 * \mathtt{z} + \mathtt{y}$$

Evaluation

$$\begin{aligned} [\![t]\!]\,\sigma &= [\![2 * \mathtt{z} + \mathtt{y}]\!]\,\{\mathtt{x} \mapsto 5, \mathtt{y} \mapsto -1, \mathtt{z} \mapsto 21\} \\ &= 2 \cdot 21 + (-1) \\ &= 41 \end{aligned}$$

This document is available free of charge on StuDocu.com
Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Proposition

For (arithmetic) expressions t, e,

$$[\![t]\!]\,(\sigma \oplus \{x \mapsto [\![e]\!]\,\sigma\}) = [\![t[e/\mathtt{x}]]\!]\,\sigma$$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Proposition

For (arithmethic) expressions t, e,

$$[\![t]\!]\,(\sigma \oplus \{x \mapsto [\![e]\!]\,\sigma\}) = [\![t[e/\mathrm{x}]]\!]\,\sigma$$

E.g.,, consider $t \equiv \mathrm{x} + \mathrm{y}$, $e \equiv 2 * \mathrm{z}$

for $\sigma = \{x \mapsto 5, y \mapsto -1, z \mapsto 21\}$.

This document is available free of charge on StuDocu.com

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Proposition

For (arithmethic) expressions t, e,

$$[\![t]\!]\,(\sigma \oplus \{x \mapsto [\![e]\!]\,\sigma\}) = [\![t[e/\mathtt{x}]]\!]\,\sigma$$

E.g.,, consider $\quad t \equiv \mathtt{x} + \mathtt{y},\ e \equiv 2 * \mathtt{z}$

for $\quad \sigma = \{x \mapsto 5, y \mapsto -1, z \mapsto 21\}$.

$$\begin{aligned}
[\![t]\!]\,(\sigma \oplus \{x \mapsto [\![e]\!]\,\sigma\}) &= [\![t]\!]\,(\sigma \oplus \{x \mapsto 42\}) \\
&= [\![t]\!]\,(\{x \mapsto 42, y \mapsto -1, z \mapsto 21\}) \\
&= 42 + (-1) \quad = \quad 41 \\
[\![t[e/\mathtt{x}]]\!]\,\sigma &= [\![(2 * \mathtt{z}) + \mathtt{y}]\!]\,\sigma \\
&= (2 \cdot 21) - 1 \quad = \quad 41
\end{aligned}$$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Proposition

$$\sigma \oplus \{x \mapsto [\![e]\!]\,\sigma\} \models t_1 < t_2 \qquad \text{iff} \qquad \sigma \models t_1[e/\mathbf{x}] < t_2[e/\mathbf{x}]$$

This document is available free of charge on StuDocu.com
Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Proposition

$$\sigma \oplus \{x \mapsto [\![e]\!]\,\sigma\} \models t_1 < t_2 \qquad \text{iff} \qquad \sigma \models t_1[e/\mathtt{x}] < t_2[e/\mathtt{x}]$$

Proof

$$\begin{array}{rcl} \sigma \oplus \{x \mapsto [\![e]\!]\,\sigma\} & \models & t_1 < t_2 \\ & \text{iff} & [\![t_1]\!]\,(\sigma \oplus \{x \mapsto [\![e]\!]\,\sigma\}) < [\![t_2]\!]\,(\sigma \oplus \{x \mapsto [\![e]\!]\,\sigma\}) \\ & \text{iff} & [\![t_1[e/\mathtt{x}]]\!]\,\sigma < [\![t_2[e/\mathtt{x}]]\!]\,\sigma \\ & \text{iff} & \sigma \models t_1[e/\mathtt{x}] < t_2[e/\mathtt{x}] \qquad \square \end{array}$$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Proposition

for every formula A,

$$\sigma \oplus \{x \mapsto [\![e]\!]\,\sigma\} \models A \quad \text{iff} \quad \sigma \models A[e/\mathbf{x}]$$

This document is available free of charge on StuDocu.com

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Proposition

for every formula A,

$$\sigma \oplus \{x \mapsto [\![e]\!]\,\sigma\} \models A \quad \text{iff} \quad \sigma \models A[e/\mathbf{x}]$$

Proof

Induction on the structure of formula A $\square$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Induction Proof of Correctness (cont.)

Step $\boxed{m > 0}$**:**

Inductive Hypothesis: The statement holds already for $m-1$.

Let B' denote the assertion at point u_{m-1}.

$\Longrightarrow \quad \sigma_{m-1} \models B'$

This document is available free of charge on StuDocu.com

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Induction Proof of Correctness (cont.)

Step $\boxed{m > 0\textbf{:}}$

Inductive Hypothesis: The statement holds already for $m - 1$.

Let B' denote the assertion at point u_{m-1}.

$\Longrightarrow \quad \sigma_{m-1} \models B'$

First, consider tests $\boxed{s_m \equiv b}$.

Then in particular, $\sigma_{m-1} = \sigma_m$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Case 1. $\sigma_m \models b$

$\Longrightarrow \quad B' \Rightarrow \mathbf{WP}[\![b]\!]\,(C, B)$ where

$\mathbf{WP}[\![b]\!]\,(C, B) \equiv (\neg b \Rightarrow C) \wedge (b \Rightarrow B)$

$\Longrightarrow \quad \sigma_m \models b \wedge (b \Rightarrow B)$

$\Longrightarrow \quad \sigma_m \models B \quad \square$

This document is available free of charge on StuDocu.com

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Case 1. $\sigma_m \models b$

$\Longrightarrow \quad B' \Rightarrow \mathbf{WP}[\![b]\!]\,(C, B)$ where

$\mathbf{WP}[\![b]\!]\,(C, B) \equiv (\neg b \Rightarrow C) \wedge (b \Rightarrow B)$

$\Longrightarrow \quad \sigma_m \models b \wedge (b \Rightarrow B)$

$\Longrightarrow \quad \sigma_m \models B \quad \square$

Case 2. $\sigma_m \models \neg b$

$\Longrightarrow \quad B' \Rightarrow \mathbf{WP}[\![b]\!]\,(B, C)$ where

$\mathbf{WP}[\![b]\!]\,(B, C) \equiv (\neg b \Rightarrow B) \wedge (b \Rightarrow C)$

$\Longrightarrow \quad \sigma_m \models \neg b \wedge (\neg b \Rightarrow B)$

$\Longrightarrow \quad \sigma_m \models B \quad \square$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Induction Proof of Correctness (cont.)

Step $\boxed{m > 0}$**:**

Induction Hypothesis: The statement holds already for $m - 1$.

Let B' denote the assertion at point u_{m-1}.

$\Longrightarrow \quad \sigma_{m-1} \models B'$

Now we deal with statements.

Case 1. $s_m \equiv \; ;$

Then
- $\sigma_{m-1} = \sigma_m$
- $\mathbf{WP}[\![;]\!]\,(B) \equiv B$

$\Longrightarrow \quad B' \Rightarrow B$

$\Longrightarrow \quad \sigma_{m-1} = \sigma_m \models B \qquad \square$

This document is available free of charge on StuDocu.com

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Case 2. $s_m \equiv$ `write(e);`

Then

- $\sigma_{m-1} = \sigma_m$
- $\mathbf{WP}[\![\texttt{;write(e)}]\!](B) \equiv B$

$\Longrightarrow \quad B' \Rightarrow B$

$\Longrightarrow \quad \sigma_{m-1} = \sigma_m \models B \quad \square$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Case 2. $s_m \equiv$ `write(e);`

Then
- $\sigma_{m-1} = \sigma_m$
- $\mathbf{WP}[\![\texttt{;write(e)}]\!]\,(B) \equiv B$

$\Longrightarrow \quad B' \Rightarrow B$

$\Longrightarrow \quad \sigma_{m-1} = \sigma_m \models B \quad \square$

Case 3. $s_m \equiv$ `x = read();`

Then
- $\sigma_m = \sigma_{m-1} \oplus \{x \mapsto c\}$ for some $c \in \mathbb{Z}$
- $\mathbf{WP}[\![\texttt{x = read();}]\!]\,(B) \equiv \forall\, x.\, B$

$\Longrightarrow \quad B' \Rightarrow \forall\, x.\, B \Rightarrow B[c/x]$

$\Longrightarrow \quad \sigma_m \models B \quad \square$

This document is available free of charge on StuDocu.com

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Case 4. $s_m \equiv$ `x = e;` Then we have:

- $\sigma_m = \sigma_{m-1} \oplus \{x \mapsto [\![e]\!]\, \sigma_{m-1}\}$
- $B' \Longrightarrow \mathbf{WP}[\![\texttt{x = e;}]\!]\,(B) \equiv B[e/x]$

$\Longrightarrow \quad \sigma_{m-1} \models B[e/x]$

$\Longrightarrow \quad \sigma_{m-1} \models B[e/x]$ iff $\sigma_m \models B$

$\Longrightarrow \quad \sigma_m \models B \qquad \square$

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Case 4. $s_m \equiv$ `x = e;` Then we have:

- $\sigma_m = \sigma_{m-1} \oplus \{x \mapsto [\![e]\!]\, \sigma_{m-1}\}$
- $B' \Longrightarrow \mathbf{WP}[\![\texttt{x = e;}]\!]\,(B) \equiv B[e/x]$

$\Longrightarrow \quad \sigma_{m-1} \models B[e/x]$

$\Longrightarrow \quad \sigma_{m-1} \models B[e/x]$ iff $\sigma_m \models B$

$\Longrightarrow \quad \sigma_m \models B \qquad \square$

This completes the proof of the theorem.

This document is available free of charge on StuDocu.com

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)

Conclusion

- The method of Floyd allows us to prove that an assertion B holds whenever (or under certain assumptions) a program point is reached ...
- For the implementation, we require:
  - the assertion **true** at the start point
  - assertions for each further program point
  - a proof that the assertions are locally consistent

  $\Longrightarrow$ Logic, automated theorem proving

Heruntergeladen durch Option Some (shuhao.zhang.x@gmail.com)