RW778 Concurrent Programming

Lecture 2: Weakest preconditions & synchronization

Last week & this week

- Last week we saw an *impossibly* complicated proof.
- This week we will look at it again, but only after learning about *weakest preconditions* — an approach that will make such proofs much easier.
- We shall start to also look at synchronization mechanisms.
- But first we need to recap two important concepts from last week.

Reminder 1

A predicate formula is equivalent to a set of states.

Example 1

Suppose that $a, b, c, d \in \{\text{true}, \text{false}\}$.

- $a \wedge b \wedge \neg c$ is equivalent to

$$\{a, b, \bar{c}, \bar{d}\} \quad \{a, b, \bar{c}, d\}$$

- $(a \Rightarrow b) \Rightarrow (c \wedge d)$ is equivalent to

$$\begin{array}{cc} \{\bar{a}, \bar{b}, c, d\} & \{a, \bar{b}, \bar{c}, \bar{d}\} \\ \{\bar{a}, b, c, d\} & \{a, \bar{b}, \bar{c}, d\} \\ \{a, b, c, d\} & \{a, \bar{b}, c, \bar{d}\} \\ & \{a, \bar{b}, c, d\} \end{array}$$

- $(a \Rightarrow \neg b) \Leftrightarrow (\neg c \wedge d)$ is equivalent to ???

Reminder 1

A predicate formula is equivalent to a set of states.

Example 2

Suppose that $x, y \in \{1, 2, 3, 4, 5\}$.

- $x = y$ is equivalent to

$$\{x{=}1, y{=}1\} \quad \{x{=}2, y{=}2\} \quad \{x{=}3, y{=}3\}$$
$$\{x{=}4, y{=}4\} \quad \{x{=}5, y{=}5\}$$

- $x + y = 5$ is equivalent to

$$\{x{=}1, y{=}4\} \quad \{x{=}2, y{=}3\} \quad \{x{=}3, y{=}2\}$$
$$\{x{=}4, y{=}1\}$$

A predicate formula is equivalent to a set of states.

Example 2

Suppose that $x, y \in \{1, 2, 3, 4, 5\}$.

- $(x \leq 3) \wedge ((x < y) \Rightarrow \mathit{isodd}(y - x))$ is equivalent to

$$\begin{array}{lll} \{x=1, y=1\} & \{x=1, y=2\} & \{x=1, y=4\} \\ \{x=2, y=1\} & \{x=2, y=2\} & \{x=2, y=3\} \\ \{x=2, y=5\} & \{x=3, y=1\} & \{x=3, y=2\} \\ \{x=3, y=3\} & \{x=3, y=4\} & \end{array}$$

- $(xy \bmod 6 = y)$ is equivalent to ???

Reminder 2

A *Hoare triple*

$$\{P\}\ S\ \{Q\}$$

(where P and Q are predicate formulas and S is a statement) is true if, whenever we start in a state that satisfies P and we execute S and it terminates, we end up in a state that satisfies Q.

Weakest preconditions

Last week we saw the following list…

- $wp(\textbf{skip}, Q) = Q$
- $wp(x := e, Q) = Q^x_e$
- $wp(v1 :=: v2, Q) = Q^{v1,v2}_{v2,v1}$
- $wp(S_1; S_2, Q) = wp(S_1, wp(S_2, Q))$
- $$\begin{aligned} wp(IF, Q) = \;& \neg(B_1 \vee \ldots \vee B_n) \Rightarrow Q \wedge \\ & (B_1 \Rightarrow wp(S_1, Q) \wedge \ldots \wedge B_n \Rightarrow wp(S_n, Q)) \end{aligned}$$
- $$\begin{aligned} wp(DO, Q) = \;& (\exists k : 0 \leq k : H_k(Q)), \text{ where} \\ & H_0(Q) = \neg BB \wedge Q \\ & H_k(Q) = H_0(Q) \vee wp(IF, H_{k-1}(Q)) \end{aligned}$$

$BB : B_1 \vee \ldots \vee B_n$

$DO : \textbf{do } BB \rightarrow IF : \textbf{if } B_1 \rightarrow S_1 \;[]\; \ldots \;[]\; B_n \rightarrow S_n \textbf{ fi do}$

…and this week we'll investigate them more closely.

$wp(\mathbf{skip}, Q) = Q$

- Trivial.

$wp(x := e, Q) = Q^x_e$

- Remember: notation Q^x_e means that we replace x by e in Q.
- $$\begin{aligned} & wp(x := 1, (x = 1)) \\ = \quad & (1 = 1) \\ = \quad & \text{true} \end{aligned}$$
- Meaning: we can start in any state (because all states satisfy true) and if we execute $x := 1$ and it terminates, we end in a state that satisfies $x = 1$.

$wp(x := e, Q) = Q^x_e$

- $$\begin{aligned} & wp(x := x + 1, (x > 0)) \\ = \quad & (x > 0)^x_{x+1} \\ = \quad & (x + 1 > 0) \\ = \quad & (x > -1) \\ = \quad & (x \geq 0) \end{aligned}$$
- Meaning: we can start in any state that satisfies $x \geq 0$ and if we execute $x := x + 1$ and it terminates, we end in a state that satisfies $x > 0$.

$wp(x := e, Q) = Q^x_e$

- $$\begin{array}{rl} & wp(x := a * b, (x = c)) \\ = & (x = c)^x_{a*b} \\ = & (a * b = c) \end{array}$$

- $$\begin{array}{rl} & wp(x := a * b, (x = b)) \\ = & ??? \end{array}$$

$wp(v1 :=: v2, Q) = Q^{v1,v2}_{v2,v1}$

- Skip.

$wp(S_1; S_2, Q) = wp(S_1, wp(S_2, Q))$

- $$\begin{aligned} & wp(i := i+1;\ j := 2*i, (j = n)) \\ = \quad & wp(i := i+1, wp(j := 2*i, (j = n))) \\ = \quad & wp(i := i+1, (j = n)^{j}_{2*i}) \\ = \quad & wp(i := i+1, (2*i = n)) \\ = \quad & (2*i = n)^{i}_{i+1} \\ = \quad & (2*(i+1) = n) \end{aligned}$$

- $$\begin{aligned} & wp(j := 2*i;\ i := i+1, (j = n)) \\ = \quad & ??? \end{aligned}$$

$wp(S_1; S_2, Q) = wp(S_1, wp(S_2, Q))$

$$
\begin{aligned}
\bullet \quad & wp(t := x;\ x := y;\ y := t, (x = Y \wedge y = X)) \\
= \quad & wp(t := x;\ x := y, wp(y := t, (x = Y \wedge y = X))) \\
= \quad & wp(t := x;\ x := y, (x = Y \wedge t = X)) \\
= \quad & wp(t := x, wp(x := y, (x = Y \wedge t = X))) \\
= \quad & wp(t := x, (y = Y \wedge t = X)) \\
= \quad & (y = Y \wedge x = X)
\end{aligned}
$$

$wp(S_1; S_2, Q) = wp(S_1, wp(S_2, Q))$

- $$wp(t := x;\ y := x;\ x := t, (x = Y \wedge y = X))$$

$wp(S_1; S_2, Q) = wp(S_1, wp(S_2, Q))$

- $$\begin{aligned} & wp(t := x;\ y := x;\ x := t, (x = Y \wedge y = X)) \\ = \quad & wp(t := x;\ y := x, wp(x := t, (x = Y \wedge y = X))) \end{aligned}$$

$wp(S_1; S_2, Q) = wp(S_1, wp(S_2, Q))$

- $$\begin{aligned} & wp(t := x;\ y := x;\ x := t, (x = Y \wedge y = X)) \\ = \quad & wp(t := x;\ y := x, wp(x := t, (x = Y \wedge y = X))) \\ = \quad & wp(t := x;\ y := x, (t = Y \wedge y = X)) \end{aligned}$$

$wp(S_1; S_2, Q) = wp(S_1, wp(S_2, Q))$

- $$\begin{aligned} & wp(t := x;\ y := x;\ x := t, (x = Y \wedge y = X)) \\ = \quad & wp(t := x;\ y := x, wp(x := t, (x = Y \wedge y = X))) \\ = \quad & wp(t := x;\ y := x, (t = Y \wedge y = X)) \\ = \quad & wp(t := x, wp(y := x, (t = Y \wedge y = X))) \end{aligned}$$

$wp(S_1; S_2, Q) = wp(S_1, wp(S_2, Q))$

- $$\begin{aligned} & wp(t := x;\ y := x;\ x := t, (x = Y \wedge y = X)) \\ = \quad & wp(t := x;\ y := x, wp(x := t, (x = Y \wedge y = X))) \\ = \quad & wp(t := x;\ y := x, (t = Y \wedge y = X)) \\ = \quad & wp(t := x, wp(y := x, (t = Y \wedge y = X))) \\ = \quad & wp(t := x, (t = Y \wedge x = X)) \end{aligned}$$

$wp(S_1; S_2, Q) = wp(S_1, wp(S_2, Q))$

- $$\begin{aligned} & wp(t := x;\ y := x;\ x := t, (x = Y \wedge y = X)) \\ = \quad & wp(t := x;\ y := x, wp(x := t, (x = Y \wedge y = X))) \\ = \quad & wp(t := x;\ y := x, (t = Y \wedge y = X)) \\ = \quad & wp(t := x, wp(y := x, (t = Y \wedge y = X))) \\ = \quad & wp(t := x, (t = Y \wedge x = X)) \\ = \quad & (x = Y \wedge x = X) \end{aligned}$$

$wp(S_1; S_2, Q) = wp(S_1, wp(S_2, Q))$

- $$\begin{aligned} & wp(t := x;\ y := x;\ x := t, (x = Y \wedge y = X)) \\ = \quad & wp(t := x;\ y := x, wp(x := t, (x = Y \wedge y = X))) \\ = \quad & wp(t := x;\ y := x, (t = Y \wedge y = X)) \\ = \quad & wp(t := x, wp(y := x, (t = Y \wedge y = X))) \\ = \quad & wp(t := x, (t = Y \wedge x = X)) \\ = \quad & (x = Y \wedge x = X) \\ = \quad & (Y = X) \end{aligned}$$

  Is this right?

$wp(IF, Q) = \ldots$

- IF : $\textbf{if}\ B_1 \rightarrow S_1\ []\ \ldots\ []\ B_n \rightarrow S_n\ \textbf{fi}$
- $wp(IF, Q) =\ \neg(B_1 \vee \ldots \vee B_n) \Rightarrow Q\ \wedge$

$$(B_1 \Rightarrow wp(S_1, Q) \wedge \ldots \wedge B_n \Rightarrow wp(S_n, Q))$$

$wp(IF, Q) = \ldots$

- **if** $x < 0 \rightarrow y := -x \;[]\; x \geq 0 \rightarrow y := x$ **fi**
- $B_1 : (x < 0) \quad S_1 : y := -x$

  $B_2 : (x \geq 0) \quad S_2 : y := x$
- $Q : y = |x|$
- $wp(IF, Q) = \neg(B_1 \vee B_2) \Rightarrow Q \wedge$

  $(B_1 \Rightarrow wp(S_1, Q) \wedge B_2 \Rightarrow wp(S_2, Q))$

$wp(IF, Q) = \ldots$

- $$\begin{aligned} & wp(S_1, Q) \\ = \quad & wp(y := -x, (y = |x|)) \\ = \quad & (-x = |x|) \\ = \quad & (x \leq 0) \end{aligned}$$

- $$\begin{aligned} & B_1 \Rightarrow wp(S_1, Q) \\ = \quad & (x < 0) \Rightarrow (x \leq 0) \\ = \quad & \text{true} \end{aligned}$$

$wp(IF, Q) = \ldots$

- $$\begin{aligned} & wp(S_2, Q) \\ = \quad & wp(y := x, (y = |x|)) \\ = \quad & (x = |x|) \\ = \quad & (x \geq 0) \end{aligned}$$

- $$\begin{aligned} & B_2 \Rightarrow wp(S_2, Q) \\ = \quad & (x \geq 0) \Rightarrow (x \geq 0) \\ = \quad & \text{true} \end{aligned}$$

$wp(IF, Q) = \ldots$

- **if** $x < 0 \rightarrow y := -x$ [] $x \geq 0 \rightarrow y := x$ **fi**

- $$\begin{aligned} wp(IF, Q) &= \neg(B_1 \vee B_2) \Rightarrow Q \wedge (B_1 \Rightarrow wp(S_1, Q) \wedge B_2 \Rightarrow wp(S_2, Q)) \\ &= \neg(B_1 \vee B_2) \Rightarrow Q \wedge (\text{true} \wedge \text{true}) \\ &= \neg(B_1 \vee B_2) \Rightarrow Q \\ &= \neg((x < 0) \vee (x \geq 0)) \Rightarrow (y = |x|) \\ &= \neg(\text{true}) \Rightarrow (y = |x|) \\ &= \text{false} \Rightarrow (y = |x|) \\ &= \text{true} \end{aligned}$$

The book says:

- DO : $\textbf{do}\ BB \rightarrow IF : \textbf{if}\ B_1 \rightarrow S_1\ [] \ \ldots\ []\ B_n \rightarrow S_n\ \textbf{fi do}$
- $BB : B_1 \vee \ldots \vee B_n$
- $wp(DO, Q) = \ (\exists k : 0 \leq k : H_k(Q))$, where

$$H_0(Q) = \neg BB \wedge Q$$
$$H_k(Q) = H_0(Q) \vee wp(IF, H_{k-1}(Q))$$

Unfortunately, the formal definition of $wp(DO, Q)$ is not easy to use, and it gives us little insight into how to develop correct software. So, we are going to use a slightly different definition.

$wp(DO, Q) = \ldots$

- DO : **do** $BB \rightarrow IF$: **if** $B_1 \rightarrow S_1$ [] $\ldots$ [] $B_n \rightarrow S_n$ **fi do**
- $BB : B_1 \vee \ldots \vee B_n$
- $\{P\}\ DO\ \{Q\}$ holds if there is an invariant I such that
  - $P \Rightarrow I$,
  - $\{I \wedge BB\}\ IF\ \{I\}$ holds, and
  - $I \wedge \neg BB \Rightarrow Q$.

$wp(DO, Q) = \ldots$

- $S: \quad x := X;\ y := Y;\ z := 0;$

  $D: \textbf{do}\ x \neq 0 \rightarrow T: x := x - 1;\ z := z + y\ \textbf{do}$
- $\{\text{true}\}\ S\ \{z = XY\}$?

$wp(DO, Q) = \ldots$

- $S: \quad x := X;\ y := Y;\ z := 0;$

  $D: \mathbf{do}\ x \neq 0 \rightarrow T: x := x - 1;\ z := z + y\ \mathbf{do}$

- $\{(x = X) \wedge (y = Y) \wedge (z = 0)\}\ D\ \{z = XY\}$?
- Where does the invariant I come from? We have to make it up!

$wp(DO, Q) = \dots$

- $S: \quad x := X;\ y := Y;\ z := 0;$

  $D : \textbf{do}\ x \neq 0 \rightarrow T : x := x - 1;\ z := z + y\ \textbf{do}$
- $\{(x = X) \wedge (y = Y) \wedge (z = 0)\}\ D\ \{z = XY\}$?
- $I = (z + xY = XY)$

$wp(DO, Q) = \ldots$

- $S: \quad x := X;\ y := Y;\ z := 0;$

  $D : \textbf{do}\ x \neq 0 \rightarrow T : x := x - 1;\ z := z + y\ \textbf{do}$
- $\{(x = X) \wedge (y = Y) \wedge (z = 0)\}\ D\ \{z = XY\}$?
- $I = (z + xY = XY)$
- We have to show
  - $P \Rightarrow I$:
  - $\{I \wedge BB\}\ T\ \{I\}$:
  - $I \wedge \neg BB \Rightarrow Q$:

$wp(DO, Q) = \ldots$

- $S: \quad x := X;\ y := Y;\ z := 0;$

  $D: \textbf{do}\ x \neq 0 \rightarrow T: x := x - 1;\ z := z + y\ \textbf{do}$

- $\{(x = X) \wedge (y = Y) \wedge (z = 0)\}\ D\ \{z = XY\}$?
- $I = (z + xY = XY)$
- We have to show
  - $P \Rightarrow I$: Trivial
  - $\{I \wedge BB\}\ T\ \{I\}$:
  - $I \wedge \neg BB \Rightarrow Q$:

$wp(DO, Q) = \ldots$

- $S: \quad x := X;\ y := Y;\ z := 0;$

  $D : \textbf{do } x \neq 0 \rightarrow T : x := x - 1;\ z := z + y \textbf{ do}$

- $\{(x = X) \wedge (y = Y) \wedge (z = 0)\}\ D\ \{z = XY\}$?

- $I = (z + xY = XY)$

- We have to show
  - $P \Rightarrow I$: Trivial
  - $\{I \wedge BB\}\ T\ \{I\}$:
  - $I \wedge \neg BB \Rightarrow Q$: Trivial

$wp(DO, Q) = \ldots$

- To show that $\{I \wedge BB\}\ T\ \{I\}$ we are going to calculate $R = wp(T, I)$ and check that $(I \wedge BB) \Rightarrow R$.

(• Recall the **Rule of Consequence:** $\dfrac{P' \Rightarrow P,\ \{P\}\ S\ \{Q\},\ Q \Rightarrow Q'}{\{P'\}\ S\ \{Q'\}}$)

- $$\begin{aligned} & wp(T, I) \\ = \quad & wp(x := x - 1;\ z := z + y, (z + xY = XY)) \\ = \quad & (z + y + xY = XY) \\ = \quad & (z + y + (x - 1)Y = XY) \\ = \quad & (z + xY = XY - y + Y) \end{aligned}$$

- Does $(I \wedge BB) \Rightarrow wp(T, I)$?

$wp(DO, Q) = \ldots$

- Does $(I \wedge BB) \Rightarrow wp(T, I)$? **NO!**
- Our invariant I is too weak. Finding the right invariant is not easy and there is no way to calculate it.
- What is missing from our invariant?

$wp(DO, Q) = \ldots$

- $I = (z + xY = XY) \wedge (y = Y)$
- Again we check
  - $P \Rightarrow I$: Trivial
  - $\{I \wedge BB\}\ T\ \{I\}$: ?
  - $I \wedge \neg BB \Rightarrow Q$: Trivial
- $wp(T, I) = I$ and because $(I \wedge BB) \Rightarrow I$, it holds that $\{I \wedge BB\}\ T\ \{I\}$.
- So, the main theorem holds: $\{\text{true}\}\ S\ \{z = XY\}$?

Answers

A predicate formula is equivalent to a set of states.

Example 1: $a, b, c, d \in \{\text{true}, \text{false}\}$.

- $(a \Rightarrow \neg b) \Leftrightarrow (\neg c \wedge d)$ is equivalent to

$$\{\bar{a}, \bar{b}, \bar{c}, d\} \quad \{\bar{a}, b, \bar{c}, d\} \quad \{a, \bar{b}, \bar{c}, d\}$$
$$\{a, b, \bar{c}, \bar{d}\} \quad \{a, b, c, \bar{d}\} \quad \{a, b, c, d\}$$

A predicate formula is equivalent to a set of states.

Example 2: $x, y \in \{1, 2, 3, 4, 5\}$.

- $(xy \bmod 6 = y)$ is equivalent to

$$\begin{array}{ccc}
\{x{=}1, y{=}1\} & \{x{=}1, y{=}2\} & \{x{=}1, y{=}3\} \\
\{x{=}1, y{=}4\} & \{x{=}1, y{=}5\} & \{x{=}3, y{=}3\} \\
\{x{=}4, y{=}2\} & \{x{=}4, y{=}4\} & \{x{=}5, y{=}3\}
\end{array}$$

Answers

- $$\begin{aligned} & wp(x := a * b, (x = b)) \\ = \quad & (x = b)^{x}_{a*b} \\ = \quad & (a * b = b) \\ = \quad & (a = 1) \vee (b = 0) \end{aligned}$$

Answers

- $$\begin{aligned} & wp(j := 2 * i;\ \ i := i + 1, (j = n)) \\ = \quad & wp(j := 2 * i, wp(i := i + 1, (j = n))) \\ = \quad & wp(j := 2 * i, (j = n)^{i}_{i+1}) \\ = \quad & wp(j := 2 * i, (j = n)) \\ = \quad & (j = n)^{j}_{2*i} \\ = \quad & (2 * i = n) \end{aligned}$$