Lecture 11
Weakest Preconditions
Review: Decorating Programs

\{true\}

x := m;
y := 0;
while (n < x) do (  
    x := x - n;
    y := y + 1
)

{  
}  

Review: Decorating Programs

\{ \text{true} \}

x := m;
y := 0;
while (n < x) do ( 
  x := x - n;
  y := y + 1
)
\{ n \times y + x = m \}

In other words, the program divides m by n, so y is the quotient and x is the remainder.
Generating Preconditions

To fill in a precondition:

\[ \{ \} \land \{ Q \} \]

there are many possible preconditions—and some are more useful than others.
Weakest Preconditions

**Intuition:** The weakest liberal precondition for $c$ and $Q$ is the weakest assertion $P$ such that $\{P\} c \{Q\}$ is valid.
Weakest Preconditions

Intuition: The weakest liberal precondition for $c$ and $Q$ is the weakest assertion $P$ such that $\{P\} c \{Q\}$ is valid.

More formally...

Definition (Weakest Liberal Precondition)

$P$ is a weakest liberal precondition of $c$ and $Q$ written $wlp(c, Q)$ if:

$$\forall \sigma, I. \sigma \vdash_I P \iff (C[c] \sigma) \text{ undefined } \lor (C[c] \sigma) \vdash_I Q$$
Weakest Preconditions

\[ wlp(\textbf{skip}, P) \triangleq P \]
Weakest Preconditions

\[
\begin{align*}
\text{wlp}(& \text{skip}, P) = P \\
\text{wlp}(& x := a, P) = P[a/x]
\end{align*}
\]

\[
\begin{align*}
(x < y + 1 & \land 2 < x+2) [\overset{\ell + d}{\xrightarrow{\ell}}] \\
= x + d < y + 1 & \land 2 < (x+2) + 2
\end{align*}
\]
Weakest Preconditions

\[
\begin{align*}
    wlp(\textbf{skip}, P) &= P \\
    wlp(x := a, P) &= P[a/x] \\
    wlp((c_1; c_2), P) &= wlp(c_1, wlp(c_2, P))
\end{align*}
\]
Weakest Preconditions

\[ wlp(\text{skip}, P) = P \]
\[ wlp(x := a, P) = P[a/x] \]
\[ wlp((c_1; c_2), P) = wlp(c_1, wlp(c_2, P)) \]
\[ wlp(\text{if } b \text{ then } c_1 \text{ else } c_2, P) = (b \implies wlp(c_1, P)) \land (\neg b \implies wlp(c_2, P)) \]

\[ P = \text{true} \]
\[ \text{false} \mid P, \neg P \]
\[ \left| P_i \Rightarrow P_2 \right| P_0 \]
Weakest Preconditions

\[
\begin{align*}
\text{wlp}(\text{skip}, P) &= P \\
\text{wlp}(x := a, P) &= P[a/x] \\
\text{wlp}((c_1; c_2), P) &= \text{wlp}(c_1, \text{wlp}(c_2, P)) \\
\text{wlp}(\text{if } b \text{ then } c_1 \text{ else } c_2, P) &= (b \implies \text{wlp}(c_1, P)) \land \\
&\quad (\neg b \implies \text{wlp}(c_2, P)) \\
\text{wlp}(\text{while } b \text{ do } c, P) &= \bigwedge_i F_i(P)
\end{align*}
\]
Weakest Preconditions

\[
\begin{align*}
\text{wlp(} \text{skip}, P \text{) } &= \ P \\
\text{wlp(} x := a, P \text{) } &= \ P[a/x] \\
\text{wlp(} (c_1; c_2), P \text{) } &= \ \text{wlp}(c_1, \text{wlp}(c_2, P)) \\
\text{wlp(} \text{if } b \text{ then } c_1 \text{ else } c_2, P \text{) } &= \ (b \implies \text{wlp}(c_1, P)) \land
\left( \neg b \implies \text{wlp}(c_2, P) \right) \\
\text{wlp(} \text{while } b \text{ do } c, P \text{) } &= \ \bigwedge_i F_i(P)
\end{align*}
\]

where

\[
\begin{align*}
F_0(P) &= \text{true} \\
F_{i+1}(P) &= \ (\neg b \implies P) \land (b \implies \text{wlp}(c, F_i(P)))
\end{align*}
\]
Applications of Weakest Preconditions

Failing fast: avoid wasting work on bad inputs.

\[
p := \text{getPacket}();
\text{processPacket}(p);
\textbf{assert } P_{\text{safe}}
\]
Applications of Weakest Preconditions

Failing fast: avoid wasting work on bad inputs.

\[
p := \text{getPacket}(); \\
\text{processPacket}(p); \\
\{ P_{\text{safe}} \}
\]
Applications of Weakest Preconditions

Failing fast: avoid wasting work on bad inputs.

\[
p := \text{getPacket}();
\{ P_{\text{filter}}(p) \};
\text{processPacket}(p);
\{ P_{\text{safe}} \} \]
Applications of Weakest Preconditions

Failing fast: avoid wasting work on bad inputs.

\[
\begin{align*}
p & := \text{getPacket}(); \\
\textbf{assert } P_{\text{filter}}(p); \\
\text{processPacket}(p); 
\end{align*}
\]
Applications of Weakest Preconditions

Failing fast: avoid wasting work on bad inputs.

\[
p := \text{getPacket}();
\]

\textbf{assert} \ P_{\text{filter}}(p);

\text{processPacket}(p);

\(P_{\text{filter}}\) should be the \textit{weakest} precondition to avoid ruling out legitimate inputs.

Properties of Weakest Preconditions

Lemma (Correctness of Weakest Preconditions)

\[ \forall c \in \text{Com}, Q \in \text{Assn}. \]
\[ \vdash \{ wlp(c, Q) \} c \{ Q \} \text{ and} \]
\[ \forall R \in \text{Assn}. \vdash \{ R \} c \{ Q \} \text{ implies } (R \implies wlp(c, Q)) \]
Properties of Weakest Preconditions

Lemma (Correctness of Weakest Preconditions)

\[ \forall c \in \textbf{Com}, Q \in \textbf{Assn}. \quad \vdash \{ \text{wlp}(c, Q) \} \ c \ \{Q\} \quad \text{and} \]
\[ \forall R \in \textbf{Assn}. \quad \vdash \{R\} \ c \ \{Q\} \implies (R \implies \text{wlp}(c, Q)) \]

Lemma (Provability of Weakest Preconditions)

\[ \forall c \in \textbf{Com}, Q \in \textbf{Assn}. \quad \vdash \{ \text{wlp}(c, Q) \} \ c \ \{Q\} \]
\[ \text{wlp} (x := a, Q) = Q[a/x] \]
Soundness and Completeness

**Soundness:** If we can prove it, then it’s actually true.

\[ \vdash \Rightarrow \models \]

**Completeness:** If it’s true, then a proof exists.

\[ \models \Rightarrow \vdash \]
Soundness and Completeness

**Soundness:** If we can prove it, then it’s actually true.

**Definition (Soundness)**

\[ \text{If } \vdash \{ P \} \triangleright \{ Q \} \text{ then } \models \{ P \} \triangleright \{ Q \}. \]

**Completeness:** If it’s true, then a proof exists.

**Definition (Completeness)**

\[ \text{If } \models \{ P \} \triangleright \{ Q \} \text{ then } \vdash \{ P \} \triangleright \{ Q \}. \]
\[ E = \{ \text{true} \} \text{ skip } \{ P \} \]
Relative Completeness

Theorem (Cook (1974))

\[ \forall P, Q \in \text{Assn}, c \in \text{Com}. \models \{P\} c \{Q\} \implies \vdash \{P\} c \{Q\}. \]

Lemma 1

Lemma 2

Conseq
Relative Completeness

Theorem (Cook (1974))

∀P, Q ∈ Assn, c ∈ Com. ⊨ \{P\} c \{Q\} implies ⊢ \{P\} c \{Q\}.

Proof Sketch.

Let \{P\} c \{Q\} be a valid partial correctness specification. By the first Lemma we have ⊨ P \implies wlp(c, Q).

By the second Lemma we have ⊢ \{wlp(c, Q)\} c \{Q\}.

We conclude ⊢ \{P\} c \{Q\} using the CONSEQUENCE rule.