Lecture 11
Weakest Preconditions
\{\texttt{true}\}\{ \\
\texttt{x := m; }\\ 
\texttt{y := 0; }\\ 
\texttt{\textbf{while} (n < x) \textbf{do} ( }\\ 
\hspace{1em} \texttt{x := x - n; }\\ 
\hspace{1em} \texttt{y := y + 1 }\\ 
\hspace{1em} ) }\\ 
\{ \\
\} \}
{true}

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

{n × 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:

\[ \{ \quad \} \ c \ \{ Q \} \]

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 $\text{wlp}(c, Q)$ if:

$$\forall \sigma, \emptyset. \sigma \models I \; P \iff (C \llbracket c \rrbracket \sigma) \text{ undefined} \lor (C \llbracket c \rrbracket \sigma) \models I \; Q$$
Weakest Preconditions

\[ wlp(\text{skip}, P) \equiv P \]
Weakest Preconditions

\[ wlp(\text{skip}, P) = P \]
\[ wlp(x := a, P) = P[a/x] \]

\[ (x < y + 1 \land z < x + 2) \]

\[ = x + 2 < y + 1 \land z < (x + 2) + 2 \]
Weakest Preconditions

\[ wlp(\text{skip}, P) = P \]
\[ wlp(x := a, P) = P[a/x] \]
\[ wlp((c_1; c_2), P) = wlp(c_1, wlp(c_2, P)) \]
Weakest Preconditions

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

\[
P = \text{true} \lor \text{false} \mid P \land P_2 \\
\mid P \Rightarrow P_2
\]
Weakest Preconditions

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

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

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.

```plaintext
p := getPacket();
processPacket(p);
assert P_{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.

\[ p := \text{getPacket}(); \]
\[ \textbf{assert} \ P_{\text{filter}}(p); \]
\[ \text{processPacket}(p); \]
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)

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

Lemma (Correctness of Weakest Preconditions)

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

Lemma (Provability of Weakest Preconditions)

\[
\forall c \in \text{Com}, Q \in \text{Assn}. \ \vdash \{\text{wlp}(c, Q)\} \ c \ \{Q\}
\]
Soundness and Completeness

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

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

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

**Definition (Soundness)**

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

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

**Definition (Completeness)**

\[ \text{If } \models \{ P \} \subset \{ Q \} \text{ then } \vdash \{ P \} \subset \{ Q \}. \]
Kurt Gödel vs. Sir Tony Hoare
Relative Completeness

Theorem (Cook (1974))

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

**Theorem (Cook (1974))**

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

**Proof Sketch.**

Let \( \{P\} c \{Q\} \) be a valid partial correctness specification.

By the first Lemma we have \( \vdash P \implies \text{wlp}(c, Q) \).

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

We conclude \( \vdash \{P\} c \{Q\} \) using the CONSEQUENCE rule.