Assertions ⊢ {P}C{Q}

⊢ {P} skip {P}

⊢ {P[e/x]}x = e{P}

⊢ {P}C1{Q} ⊢ {Q}C2{R}
⊥ {P}C1; C2{R}

⊢ {P and B}C1{Q} ⊢ {P and not B}C2{Q}
⊥ {P} if B then C1 else C2{Q}

⊢ P1 implies P2 ⊢ {P2}C{Q2} ⊢ Q2 implies Q1
⊥ {P1}C{Q1}
⊢ \{P \text{ and } B\} C\{Q\}

⊢ \{P\} \text{while } B \text{ do } C\{Q \text{ and not } B\}

⊢ \{P \text{ and } B\} C\{P\}

⊢ \{P\} \text{while } B \text{ do } C\{P \text{ and not } B\}

⊢ \{P\} C\{Q \text{ and not } B\}

⊢ \{P\} \text{while } B \text{ do } C\{Q \text{ and not } B\}

⊢ \{P\} C\{P\}

⊢ \{P\} \text{while } B \text{ do } C\{P \text{ and not } B\}

P is loop invariant
\( s \in S : \text{Id} \rightarrow \text{Val} \)

\[ [C] : S \rightarrow S \cup \{\perp\} \]

\[
[\text{skip}] s = s \quad [x = e] s = s\{x \rightarrow [e] s\} \quad [C_1; C_2] s = [C_2][C_1] s
\]

\[
[\text{if } B \text{ then } C_1 \text{ else } C_2] s = \text{if } [B] s \text{ then } [C_1] s \text{ else } [C_2] s
\]

\[
[\text{while } B \text{ do } C] s = \text{if not } [B] s \text{ then } s \text{ else } [\text{while } B \text{ do } C][C] s
\]

\[
[\text{while } B \text{ do } C] s = \text{fix}(\lambda f (\lambda s \text{ if not } [B] s \text{ then } s \text{ else } f [C] s))
\]
\([E] : S \rightarrow Val\)

\([n]_s = n\)

\([v]_s = s(v)\)

\([E1 + E2]_s = [E1]_s + [E2]_s\)

\([E1 \ast E2]_s = [E1]_s \ast [E2]_s\)
\([B]s : S \rightarrow \text{Bool}\)

\([B1 \text{ and } B2]s = [B1]s \text{ and } [B2]s\)
\([B1 \text{ or } B2]s = [B1]s \text{ or } [B2]s\)
\([\text{not } B]s = \text{not } [B]s\)

\([E1 \leq E2]s = [E1]s \leq [E2]s\)
\([E1 = E2]s = [E1]s = [E2]s\)

\([\text{all } j.B]s = \forall v \in \text{Val}. [B]s\{j \rightarrow v\}\)
\([\text{exists } j.B]s = \exists v \in \text{Val}. [B]s\{j \rightarrow v\}\)
Proof: $\vdash \{P\} C \{Q\}$ if can prove using logic

Validity: $\models \{P\} C \{Q\} \equiv \forall s. [P]_s \implies [Q][C]_s$

Consistency: $\vdash \{P\} C \{Q\}$ implies $\models \{P\} C \{Q\}$

Completeness: $\models \{P\} C \{Q\}$ implies $\vdash \{P\} C \{Q\}$
Assume $\models \{P\}C1; C2\{Q\}$

Want $\vdash \{P\}C1; C2\{Q\}$

Must find $R$ such that

• $\vdash \{P\}C1\{R\}$

• $\vdash \{R\}C2\{Q\}$
Assume $|= \{P\}$ while $B$ do $C\{Q\}$

Want $\vdash \{P\}$ while $B$ do $C\{Q\}$

Must find $R$ such that

- $\vdash \{R\}$ while $B$ do $C\{R \text{ and not } B\}$

- $P$ implies $R$

- $R$ and not $B$ implies $Q$
• Why do we believe such an $R$ exists?

• If it does exist, how can we find it?

• If we can find it, can we prove the necessary implications?
Weakest Preconditions

\( wp(C\{Q\}) = \{s.\llbracket C \rrbracket s = \bot \text{ or } \llbracket Q \rrbracket \llbracket C \rrbracket s\} \)

Given \( C\{Q\} \), want \( R \) such that \( \llbracket R \rrbracket s \text{ iff } s \in wp(C\{Q\}) \)

\( R \) is the \textit{weakest precondition} \( wpc(C\{Q\}) \) of \( C\{Q\} \)

To prove \( \{P\}C\{Q\} \):

- Find weakest precondition \( R \) of \( C\{Q\} \)  
  (potential problem - can’t express \( R \) in logic)

- Prove \( P \) implies \( R \)  
  (potential problem - \( P \) implies \( R \) valid but unprovable)
Computing Weakest Preconditions

\[ \text{wpc(skip \ \{Q\})} = Q \]

\[ \text{wpc}(x = e\{Q\}) = Q[e/x] \]

\[ \text{wpc}(C_1; C_2\{Q\}) = \text{wpc}(C_1\{\text{wpc}(C_2\{Q\})\}) \]

\[ \text{wpc}(\text{if } B \text{ then } C_1 \text{ else } C_2\{Q\}) = (B \text{ and } \text{wpc}(C_1\{Q\})) \text{ or } (\text{not } B \text{ and } \text{wpc}(C_2\{Q\})) \]

Assertion language must be closed under substitution, conjunction, disjunction, and negation.
Verification Condition Generation

Given \( \{P\}C[I]\{Q\} \)

- Assertion \( P \) at start of code (precondition)
- Assertion \( Q \) at end of code (postcondition)
- Assertion \( I \) for each loop (loop invariant)

Use weakest precondition rules to generate verification condition.

Verification condition is logical assertion. If verification condition is true, then \( \vdash \{P\}C\{Q\} \).
Verification Condition Generation Example

\{0 \leq N\}
\begin{align*}
i &= 0; f = 1; \\
{0 \leq i \leq N \text{ and } f = i!}\end{align*}
\begin{align*}
\text{while } i \neq N \text{ do } i &= i + 1; f = f \cdot i \\
\{f = N!\}\end{align*}

Use goal-directed reasoning
Goal 1: Verify loop invariant
Step 1: Extract loop body
Step 2: Place loop invariant at end (prepare to apply while loop rule)

\{0 \leq i \leq N \text{ and } f = i! \text{ and } i \neq N\}
i = i + 1; f = f \ast i
\{0 \leq i \leq N \text{ and } f = i!\}
Goal 2: Push loop invariant back through loop body
Step 1: Derive intermediate assertions (wpc)
Step 2: Extract verification condition (consequence)

\{0 \leq i \leq N \text{ and } f = i! \text{ and } i \neq N\}
\{0 \leq i + 1 \leq N \text{ and } f \ast (i + 1) = (i + 1)!\}
i = i + 1;
\{0 \leq i \leq N \text{ and } f \ast i = i!\}
f = f \ast i
\{0 \leq i \leq N \text{ and } f = i!\}

Verification condition:
\{0 \leq i \leq N \text{ and } f = i! \text{ and } i \neq N\}
implies
\{0 \leq i + 1 \leq N \text{ and } f \ast (i + 1) = (i + 1)!!\}
Goal 3: Push loop invariant back to precondition
Step 1: Derive intermediate assertions (wpc)
Step 2: Extract verification condition (consequence)
\[ \{0 \leq N\} \]
\[ \{0 \leq 0 \leq N \text{ and } 1 = 0!\} \]
i = 0;
\[ \{0 \leq i \leq N \text{ and } 1 = i!\} \]
f = 1;
\[ \{0 \leq i \leq N \text{ and } f = i!\} \]

Verification condition:
\[ \{0 \leq N\} \text{ implies } \{0 \leq 0 \leq N \text{ and } 1 = 0!\} \]
Goal 4: Push postcondition back to loop
Step 1: Extract verification condition (consequence)

\{0 \leq i \leq N \text{ and } f = i!\}
while \( i \neq N \) do \( i = i + 1; \quad f = f \times i \)
\{0 \leq i \leq N \text{ and } f = i! \text{ and not } i \neq N\}
\{f = N!\}

Verification condition:
\{0 \leq i \leq N \text{ and } f = i! \text{ and not } i \neq N\} \text{ implies } \{f = N!\}
Goal 5: Verify Verification Conditions

\{0 \leq i \leq N \text{ and } f = i! \text{ and } i \neq N\} \\
\text{implies} \\
\{0 \leq i + 1 \leq N \text{ and } f \ast (i + 1) = (i + 1)!\}

\{0 \leq N\} \text{ implies } \{0 \leq 0 \leq N \text{ and } 1 = 0!\}

\{0 \leq i \leq N \text{ and } f = i! \text{ and not } i \neq N\} \text{ implies } \{f = N!\}
Strongest Postconditions

\[ \text{sp}(\{P\}C) = \{\llbracket C \rrbracket s, \llbracket P \rrbracket s\} - \{\bot\} \]

Given \( \{P\}C \), want \( R \) such that \( \llbracket R \rrbracket s \iff s \in \text{sp}(\{P\}C) \)

\( R \) is the strongest postcondition \( \text{spc}(\{P\}C) \) of \( \{P\}C \)

To prove \( \{P\}C\{Q\} \):

- Find strongest postcondition \( R \) of \( \{P\}C \)
  (potential problem - can’t express \( R \) in logic)

- Prove \( R \) implies \( Q \)
  (potential problem - \( R \) implies \( Q \) valid but unprovable)
Computing Strongest Postconditions

\[ \text{spc}(\{P\}\text{skip}) = P \]

\[ \text{spc}(\{P\}x = e) = \exists y. x = e[y/x] \text{ and } P[y/x] \]

\[ \text{spc}(\{P\}C_1; C_2) = \text{spc}(\{\text{spc}(\{P\}C_1)\}C_2) \]

\[ \text{spc}(\{P\}\text{if } B \text{ then } C_1 \text{ else } C_2) = \text{spc}(\{P \text{ and } B\}C_1) \text{ or } \text{spc}(\{P \text{ and not } B\}C_2) \]

Assertion language must be closed under existential quantification, substitution, conjunction, disjunction, and negation.
Goal 1: Verify loop invariant
Step 1: Extract loop body
Step 2: Place loop invariant at end (prepare to apply while loop rule)

\{0 \leq i \leq N \text{ and } f = i! \text{ and } i \neq N\}
i = i + 1; f = f \ast i
\{0 \leq i \leq N \text{ and } f = i!\}
Goal 2: Push loop invariant forward through loop body  
Step 1: Derive intermediate assertions (spc)  
Step 2: Extract verification condition (consequence)  

\{0 \leq i \leq N \text{ and } f = i! \text{ and } i \neq N\}  
i = i + 1;  
\{\exists y. i = y + 1 \text{ and } 0 \leq y \leq N \text{ and } f = y! \text{ and } y \neq N\}  
f = f \ast i  
\{\exists z. f = z \ast i \text{ and } \exists y. i = y + 1  
\text{ and } 0 \leq y \leq N \text{ and } z = y! \text{ and } y \neq N\}  
\{0 \leq i \leq N \text{ and } f = i!\}
Verification condition:
{\exists z. f = z \times i \text{ and } \exists y. i = y + 1
 \text{ and } 0 \leq y \leq N \text{ and } z = y! \text{ and } y \neq N}\}
implies
{0 \leq i \leq N \text{ and } f = i!}\}
Goal 3: Push precondition forward to loop invariant
Step 1: Derive intermediate assertions (spc)
Step 2: Extract verification condition (consequence)
\{0 \leq N\}
\(i = 0;\)
\{\exists y. x = 0 \text{ and } 0 \leq N\}
\(f = 1;\)
\{\exists z. f = 1 \text{ and } \exists y. i = 0 \text{ and } 0 \leq N\}
\{0 \leq i \leq N \text{ and } f = i!\}

Verification condition:
\{\exists z. f = 1 \text{ and } \exists y. x = 0 \text{ and } 0 \leq N\}
implies
\{0 \leq i \leq N \text{ and } f = i!\}
Goal 4: Push loop invariant to postcondition
Step 1: Extract verification condition (consequence)

\[
\{0 \leq i \leq N \text{ and } f = i!\}
\]
while \( i \neq N \) do \( i = i + 1; f = f \times i \)
\[
\{0 \leq i \leq N \text{ and } f = i! \text{ and not } i \neq N\}
\]
\[
\{f = N!\}
\]

Verification condition:
\[
\{0 \leq i \leq N \text{ and } f = i! \text{ and not } i \neq N\} \text{ implies } \{f = N!\}
Goal 5: Verify Verification Conditions

Verification condition:
\[
\{ \exists z. f = z \ast i \quad \text{and} \quad \exists y. i = y + 1 \\
\text{and} \quad 0 \leq y \leq N \quad \text{and} \quad z = y! \quad \text{and} \quad y \neq N \}
\]
implies
\[
\{ 0 \leq i \leq N \quad \text{and} \quad f = i! \}
\]

Verification condition:
\[
\{ \exists z. f = 1 \quad \text{and} \quad \exists y. i = 0 \quad \text{and} \quad 0 \leq N \}
\]
implies
\[
\{ 0 \leq i \leq N \quad \text{and} \quad f = i! \}
\]

Verification condition:
\[
\{ 0 \leq i \leq N \quad \text{and} \quad f = i! \quad \text{and} \quad \text{not} \ i \neq N \}
\]
implies
\[
\{ f = N! \}