Program Correctness (Section 8.1)
Program Correctness (for imperative programs)

- A theory of program correctness needs wffs, axioms, and inference rules. The wffs (called Hoare triples) are of the form: \( \{ P \} \ S \{ Q \} \) where \( S \) is a program statement and \( P \) (a precondition) and \( Q \) (a postcondition) are logical statements about the variables of \( S \).

- **Semantics:** The meaning of \( \{ P \} \ S \{ Q \} \) is the truth value of the statement:
  - If \( P \) is true before \( S \) executes, then \( Q \) is true after \( S \) halts.

- Note that it is assumed that \( S \) halts. If \( \{ P \} \ S \{ Q \} \) is true, then \( S \) is said to be *correct* with respect to (wrt) precondition \( P \) and postcondition \( Q \).
• The Assignment Axiom (AA) is: $\{P(x/t)\} \ x := t \{P\}$.

• Example: $\{x = 4\} \ x := x - 1 \{x = 3\}$

• Consequence Rules are
  - $P \rightarrow R$ and $\{R\}S\{Q\}$
    - $\{P\}S\{Q\}$
  - $\{P\}S\{T\}$ and $T \rightarrow Q$
    - $\{P\}S\{Q\}$
Example Proof

Prove the correctness of $\{x < 3\} x := x - 1 \{x < 3\}$. Proof:

1. $\{x < 4\} x := x - 1 \{x < 3\}$  AA
2. $x < 3$  P [for $(x < 3) \rightarrow (x < 4)$]
3. $x < 4$  2,T
4. $(x < 3) \rightarrow (x < 4)$  2,3,CP
5. $\{x < 3\} x := x - 1 \{x < 3\}$  1,4,Consequence.  QED
Another Correctness Proof

Prove the correctness of
\{\exists x \ (y = 2x)\} y := y + 3\{\exists x \ (y = 2x + 1)\}.

1. \{\exists x \ (y + 3 = 2x + 1)\} y := y + 3
\{\exists x \ (y = 2x + 1)\}  
   AA
2. \exists x \ (y = 2x)  
   P [for CP]
3. y = 2c  
   2, EI
4. (y = 2c) \rightarrow (y + 3 = 2c + 3)  
   EE, where f(x) = x + 3
5. y + 3 = 2c + 3  
   3,4, MP
6. y + 3 = 2(c + 1) + 1  
   5, T
7. \exists x \ (y + 3 = 2x + 1)  
   6, EG
8. \exists x \ (y = 2x) \rightarrow \exists x \ (y + 3 = 2x + 1)  
   2-7, CP
9. \{\exists x \ (y = 2x)\} y := y + 3\{\exists x \ (y = 2x + 1)\}  
   1,8, Consequence
QED
Composition Rule

• Composition Rule is:
  \( \{P\}S_1\{Q\} \text{ and } \{Q\}S_2\{R\} \)
  \( \{P\}S_1;S_2\{R\} \)
Example proof

Prove the correctness of \( \{ x < 2 \} y := 2x; x := y - 3 \{ x < 1 \} \)

1. \( \{ y - 3 < 1 \} x := y - 3 \{ x < 1 \} \)  \( \text{AA} \)
2. \( \{ 2x - 3 < 1 \} y := 2x \{ y - 3 < 1 \} \)  \( \text{AA} \)
3. \( \{ 2x - 3 < 1 \} y := 2x; x := y - 3 \{ x < 1 \} \)  \( 1,2,\text{Composition} \)
4. \( x < 2 \)  \( \text{P} \text{ [for } (x < 2) \rightarrow (2x - 3 < 1) \text{]} \)
5. \( 2x < 4 \)  \( 4,\text{T} \)
6. \( 2x - 3 < 1 \)  \( 5,\text{T} \)
7. \( (x < 2) \rightarrow (2x - 3 < 1) \)  \( 4-6,\text{CP} \)
8. \( \{ x < 2 \} y := 2x; x := y - 3 \{ x < 1 \} \)  \( 3,7,\text{Consequence} \)

QED
If Rules

- **If-Then Rule:**
  \[ \{ P \land C \} S \{ Q \} \text{ and } P \land \neg C \rightarrow Q \]
  \[ \{ P \} \text{ if } C \text{ then } S \{ Q \} \]

- **If-Then-Else Rule:**
  \[ \{ P \land C \} S_1 \{ Q \} \text{ and } \{ P \land \neg C \} S_2 \{ Q \} \]
  \[ \{ P \} \text{ if } C \text{ then } S_1 \text{ else } S_2 \{ Q \} \]
If proof

Prove the correctness of \( \{x > 0\} \text{if } x > 1 \text{ then } x := x - 1 \{x > 0\} \)

1. \( \{x - 1 > 0\} x := x - 1 \{x > 0\} \)  
2. \((x > 0) \land (x > 1)\)  
3. \(x > 1\)  
4. \(x - 1 > 0\)  
5. \((x > 0) \land (x > 1) \rightarrow (x - 1 > 0)\)  
6. \(\{(x > 0) \land (x > 1)\} x := x - 1 \{x > 0\}\)  
7. \((x > 0) \land \neg(x > 1)\)  
8. \(x > 0\)  
9. \((x > 0) \land \neg(x > 1) \rightarrow (x > 0)\)  
10. \(\{x > 0\} \text{if } x > 1 \text{ then } x := x - 1 \{x > 0\}\)

QED
While rule

- \(
\{P \land C\} S\{P\}
\)

- \(
\{P\} \text{while } C \text{ do } S\{P \land \neg C\}
\)

- \(P\) is called a \emph{loop invariant}
Additional issues

Two additional issues that the book investigates are: arrays and array indices, and loop termination.
This section is about Higher-Order Logic. This allows us to, for example, quantify over predicates, rather than just variables. We won’t study this idea.