PROGRAM CORRECTNESS

PROGRAM = CODE + CORRECTNESS PROOF
          (VERIFICATION)

SYNTACTICAL CORRECTNESS = THE FORMAT IS OK
                         EASY TO VERIFY, EVERY COMPILER DOES IT

SEMANTICAL CORRECTNESS = THE PROGRAM TERMINATES
                        AND GIVES THE CORRECT OUTPUT

UNIVERSAL AUTOMATED VERIFICATION IS
IMPOSSIBLE. DOES NOT FOLLOW FROM
THE SYNTACTICAL CORRECTNESS

CONSTRUCTIVE REASONING:  CORRECTNESS PROOF
                     YIELDS A CODE OF P

CORRECT PROGRAM = CONSTRUCTIVE PROOF
                  OF IS CORRECTNESS

ITERATIVE PROGRAM = PROOF BY MATHEMATICAL
                    INDUCTION
TH (RICE). NO NONTRIVIAL PROPERTY OF A
COMPUTABLE FUNCTION CAN BE
AUTOMATICALLY VERIFIED FROM ITS
PROGRAM.

SPECIAL CASE – HALTING PROBLEM

\[ P_i = \text{THE PROGRAM WITH THE CODE } i \]

\[ \text{PROGRAM CODE } i \]

\[ \text{OPERATING SYSTEM } \Phi \]

\[ \text{PROGRAM INPUT } x \]

\[ \text{OUTPUT } \Phi(i, x) = P_i(x) \]

TH THERE IS NO ALGORITHM OR SUCH THAT

\[ O_\tau(i, x) = \begin{cases} 
1 & \text{if } P_i(x) \text{ TERMINATES} \\
0 & \text{otherwise}
\end{cases} \]

PROOF. SUPPOSE SUCH AN ALGORITHM EXISTS. DEFINE
AN AUXILIARY FUNCTION

\[ B(m) = \begin{cases} 
0 & \text{if } O_\tau(m, m) = 1 \land \Phi(m, m) \neq 0 \\
1 & \text{if } O_\tau(m, m) = 0 \land (O_\tau(m, m) = 1 \land \Phi(m, m) = 0)
\end{cases} \]

TOTAL, COMPUTABLE

LET \( x \) BE A CODE OF \( B(y) \), i.e. \( B(k) = \Phi(n, k) \) FOR
NOTE THAT \( O_\tau(n, k) = 1 \), SINCE \( B \) IS TOTAL, ALL \( k \)'s
\( B(n) = 0 \Rightarrow \Phi(n, n) \neq 0 \Rightarrow B(n) \neq 0 \)
\( B(n) \neq 0 \Rightarrow B(n) = 1 \Rightarrow \Phi(n, n) = 0 \Rightarrow B(n) = 0 \)

CONTRADICTION!
A program $P$ is said to be **partially correct** if the correct output is obtained whenever $P$ terminates.

**Proving partial correctness**

**Hoare Implication**

$P \{ S \} q$

- **Initial assertion**
- **Program segment**
- **Final assertion**

$P \{ S \} q \iff$ if $P$ is true for the input value(s) and $S$ terminates then $q$ is true for the output value(s).

**Example**

$S$: \[
\begin{align*}
    y &:= 2 \\
    z &:= x + y \\
    2 &:= x + y \\
    q &:= 2 = 3
\end{align*}
\]

$P \{ S \} q$ holds. Indeed, assuming $x = 1$ we have $y = 2$ and $z = x + y = 1 + 2 = 3$.

**Terminology:** $S$ is (partially) correct with respect to the initial assertion $P$ and the final assertion $q$. 

-104-
RULES OF INFERENCE FOR HOARE IMPLICATION

THE COMPOSITION RULE

\[ \frac{p \{ S_1 \} q}{q \{ S_2 \} r} \quad \frac{p \{ S_2 \} r}{p \{ S_1 ; S_2 \} r} \]

S = S_1 ; S_2 is S_1, followed by S_2

THE CONDITIONAL RULE

\[
\begin{align*}
(p \text{A CONDITION}) \{ S \} q & \quad \text{IF CONDITION THEN } \{ S \} q \\
(p \text{A T-CONDITION}) \rightarrow q & \quad \text{IF CONDITION IS TRUE THEN EXECUTE } S \text{. D.W. DON'T.} \\
(p \text{A CONDITION}) \{ S_1 \} q & \quad \text{IF CONDITION THEN } S_1 \\
(p \text{A T-CONDITION}) \{ S_2 \} q & \quad \text{ELSE } S_2 \\
p \{ \text{IF CONDITION THEN } S_1, \text{ ELSE } S_2 \} q & \quad \text{-105-}
\end{align*}
\]
EXAMPLE

VERIFY THAT THE SEGMENT IS

\[
\begin{align*}
\text{IF } x < 0 \text{ THEN } & \quad \text{abs} := -x \\
\text{ELSE } & \quad \text{abs} := x
\end{align*}
\]

CORRECT W.R.T. THE

INITIAL ASSERTION \( T \)

AND THE FINAL ASSERTION \( \text{abs} = |x| \)

\( T \) IS A DEFAULT INITIAL ASSERTION "ASSUME

NOTHING PARTICULAR"

\[
\begin{align*}
1^\circ \quad x < 0 & \Rightarrow \text{abs} = -x = |x| \\
2^\circ \quad x \geq 0 & \Rightarrow \text{abs} = x = |x|
\end{align*}
\]

\[
(x < 0) \lor (x \geq 0) \Rightarrow \text{abs} = |x|
\]

LOOP INVARIANTS, CONSIDER A SEGMENT

\[
\text{WHILE } \text{CONDITION} \quad S
\]

S IS REPEATEDLY EXECUTED
UNTILL CONDITION BECOMES
FALSE

\( P \) IS A LOOP INVARIANT IF \( (P \land \text{CONDITION}) \{ S \} P \)

RULE OF INFERENCE:

\[
(P \land \text{CONDITION}) \{ S \} P
\]

\( P \text{ \& WHILE CONDITION } S \{ (\neg \text{CONDITION A} P) \)
Example: Verify that the segment produces \( \text{factorial} = n! \)

\[
\begin{align*}
i_1 &= 1 \\
\text{factorial} &= 1 \\
\text{WHILE } i \leq n \\
\text{BEGIN} \\
\quad i &= i + 1 \\
\quad \text{factorial} &= \text{factorial} \times i \\
\text{END}
\end{align*}
\]

Solution. 1) Check that \( p(i) \) is a loop invariant

\( p(i) \): "factorial = \( i! \) \land i \leq n"

Induction on \( i \):

\[
(p(i) \land \text{condition}(i)) \land (3) \implies p(i+1) = A(i)
\]

\[
A(i) \left\{ \begin{array}{l}
p(i): \text{factorial} = 1 \land 1 \leq n \land 1 \leq i \\
p(2): \text{factorial} = 2 \land 2 \leq n \\
\end{array} \right.
\]

\[
i = k \left\{ \begin{array}{l}
\text{factorial} = k! \land k \leq n \land k \leq n \\
p(k): \text{factorial} = (k+1)! \land k \leq n \land p(k+1)
\end{array} \right.
\]

\[A(k) \Rightarrow A(k+1)\]

2) Conclude: \( p(\text{WHILE } i < n \text{ S} \text{[i > n \land p]} \text{)} \imsieq i = n \implies p(n) \implies \text{factorial} = n! \)

Kw 3.5: 2, 4, 10