LECT 16  10/6/99

PROGRAM CORRECTNESS

PROGRAM IS CODE + CORRECTNESS PROOF (VERIFICATION)

SYNTactical CORRECTNESS = THE PROGRAM IS OR 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 OR P

CORRECT PROGRAM = CONSTRUCTIVE PROOF OF IS CORRECTNESS

ITERATIVE PROGRAM = PROOF BY MATHEMATICAL INDUCTION

A PROGRAM P IS SAID TO BE PARTIALLY CORRECT IF THE CORRECT OUTPUT IS OBTAINED WHENEVER P TERMINATES.

PROVING PARTIAL CORRECTNESS

HOARE IMPERATION: P (S, q)

INITIAL ASSERTION

PROGRAM SEGMENT

FINAL ASSERTION

P (S)q <== IF P IS TRUE FOR THE INPUT VALUES AND S TERMINATES THEN q IS TRUE FOR THE OUTPUT VALUES.

EXAMPLE

S:

\[
\begin{align*}
&y = 2, \\
&z = x + y
\end{align*}
\]

q: z = 2

P (S)q HOLDs, 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.

TH(G)E. NO NONTRIVIAL PROPERTY OF A COMPUTABLE FUNCTION CAN BE AUTOMATICALLY VERIFIED FROM ITS PROGRAM.

SPECIAL CASE - HALTING PROBLEM

PROGRAM CODE L

PROGRAM INPUT X

OPERATING SYSTEM

OUTPUT P(x)

TH THERE IS NO ALGORITHM ORX SUCH THAT

\[
O(x) = \begin{cases} 
1, & \text{if } P(x) \text{ TERMINATES} \\
0, & \text{o.w.}
\end{cases}
\]

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

B(m) = \begin{cases} 
0, & \text{if } O(m) = 1 \land \Phi(m) \neq 0 \\
1, & \text{if } O(m) = 0 \lor \Phi(m) = 0
\end{cases}

LET M BE A CODE OF B(m), i.e., B(m) = \Phi(m). FOR NOTE THAT B(m) = 1, SINCE B IS TOTAL.

B(m) = 0 \Rightarrow \Phi(m) \neq 0 \Rightarrow B(m) = 0 \Rightarrow \Phi(m) = 0 \Rightarrow B(m) = 0

CONTRADITION!

RULES OF INFERENCE FOR HOARE IMPERATION

THE COMPOSITION RULE

P (S, q)

P (S, S)

S = S, S IS S, FOLLOWED BY S

THE CONDITIONAL RULE COVERS A PROGRAM SEGMENT

(P \land \text{CONDITION}) \Rightarrow q

IF CONDITION THEN S

(P \land \text{CONDITION}) \Rightarrow q

(P \land \text{CONDITION}) \Rightarrow q

IF CONDITION THEN S, ELSE S

(P \land \text{CONDITION}) \Rightarrow q

(P \land \text{CONDITION}) \Rightarrow q

IF CONDITION THEN S, ELSE S

103
EXAMPLE: VERIFY THAT THE SEGMENT IS CORRECT W.R.T. THE INITIAL ASSERTION $T$! AND THE FINAL ASSERTION $abs = |x|$

$T$ is a default initial assertion "assume nothing particular"

1. $x < 0$ \[ \Rightarrow \] $abs = -x = |x|$
2. $x \geq 0$ \[ \Rightarrow \] $abs = x = |x|$

\[ (x < 0) \lor (x = 0) \Rightarrow abs = |x| \]

**Loop Invariants**

Consider a segment

While condition $S$ is repeatedly executed until condition becomes false

$P$ is a loop invariant if $(P \land \text{condition}) \{ S \} P$

Rule of Inference:

$(P \land \text{condition}) \{ S \} P$

$P \land \text{condition} \{ S \} (\text{condition} \land P)$

---

EXAMPLE: VERIFY THAT THE SEGMENT PRODUCES $\text{factorial} = n!$.

\[ i := 1 \]
\[ \text{factorial} := 1 \]
\[ \text{while } i \leq n \]
\[ \begin{align*}
\text{i} & := i + 1 \\
\text{factorial} & := \text{factorial} \times i
\end{align*} \]
\[ \text{end} \]

Solution: Check that $P(i)$ is a loop invariant

$P(i): \ "\text{factorial} = i! \land i \leq n"$

induction on $i$, $(P(i) \land \text{condition}(i)) \{ S \} P(i + 1) = A(i)$

\[ i = 1 \]
\[ P(1) \land \text{condition}(1): \text{factorial} = 1 \land 1 \leq 1 \leq n \]

\[ A(i) = \{ \]
\[ P(i): \ "\text{factorial} = i! \land i \leq n" \]
\[ \text{in} \{ \text{factorial} = n! \land n \leq n \land \text{condition}(n) \}
\[ A(n): \ "\text{factorial} = n! \land n \leq n \land \text{condition}(n) \}
\[ 2) \text{conclude: } P(\text{while } i \leq n) \{ S \} (\text{condition} \land P)$

\[ \text{HW 3.5: } 2, 4, 10 \]