LEC.16 10/6/99

PROGRAM CORRECTNESS

PROGRAM IS 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

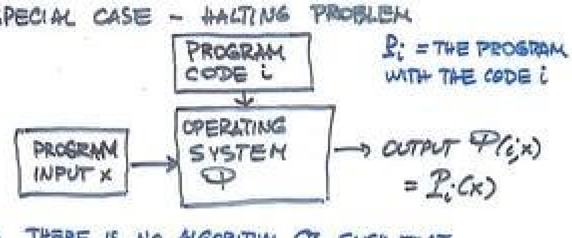
CONSTRUCTIVE REASONING: CORRECTNESS PROOF YIELDS A CODE OF P
(R.CONSTABLE, et al.)

CORRECT PROGRAM $\equiv$ 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



Th. THERE IS NO ALGORITHM $\mathfrak{A}$ SUCH THAT

$$\mathfrak{A}(i,x) = \begin{cases} 1, & \text{IF } P_i(x) \text{ TERMINATES} \\ 0, & \text{O.W.} \end{cases}$$

PROOF: SUPPOSE SUCH ALGORITHM EXISTS. DEFINE AN AUXILIARY FUNCTION

$$B(m) = \begin{cases} 0, & \text{IF } \mathfrak{A}(m,m)=1 \wedge \varphi(m,m)\neq 0 \\ 1, & \text{IF } \mathfrak{A}(m,m)=0 \vee (\mathfrak{A}(m,m)=1 \wedge \varphi(m,m)=0) \end{cases}$$

TOTAL, COMPUTABLE

LET n BE A CODE OF $B(y)$, i.e. $B(k)=\varphi(n,k)$ FOR ALL k'S
NOTE THAT $\mathfrak{A}(n,k)=1$, SINCE B IS TOTAL.
$B(n)=0 \Rightarrow \varphi(n,n)\neq 0 \Rightarrow B(n)\neq 0$
$B(n)\neq 0 \Rightarrow B(n)=1 \Rightarrow \varphi(n,n)=0 \Rightarrow B(n)=0$ 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 (p), PROGRAM SEGMENT (S), FINAL ASSERTION (q)

$p\{S\}q \iff$ IF p IS TRUE FOR THE INPUT VALUE(S) AND S TERMINATES THEN q IS TRUE FOR THE OUTPUT VALUE(S).

EXAMPLE $S: \begin{cases} y := 2 \\ z := x+y \end{cases}$ $p: x=1$ $q: z=3$

$p\{S\}q$ 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

RULES OF INFERENCE FOR HOARE IMPLICATION

THE COMPOSITION RULE

$$\frac{p\{S_1\}q \qquad q\{S_2\}r}{p\{S_1;S_2\}r}$$

$S = S_1;S_2$ IS S_1 FOLLOWED BY S_2

THE CONDITIONAL RULE COVERS A PROGRAM SEGMENT

$$\frac{(p \wedge \text{CONDITION})\{S\}q \qquad (p \wedge \neg\text{CONDITION}) \rightarrow q}{p\{\text{IF CONDITION THEN } S\}q}$$

IF CONDITION THEN
S

IF CONDITION IS TRUE THEN EXECUTE S. O.W. DON'T.

$$\frac{(p \wedge \text{CONDITION})\{S_1\}q \qquad (p \wedge \neg\text{CONDITION})\{S_2\}q}{p\{\text{IF CONDITION THEN } S_1 \text{ ELSE } S_2\}q}$$

IF CONDITION THEN
S_1
ELSE
S_2

EXAMPLE. VERIFY THAT THE SEGMENT IS

```
IF x<0 THEN
      abs := -x
ELSE
      abs := x
```

CORRECT W.R.T. THE INITIAL ASSERTION T AND THE FINAL ASSERTION $abs = |x|$

(T IS A DEFAULT INITIAL ASSERTION "ASSUME NOTHING PARTICULAR")

1° $x < 0 \implies abs = -x = |x|$

2° $x \geq 0 \implies abs = x = |x|$

$\underbrace{(x<0) \vee (x \geq 0)}_{T} \implies abs = |x|$

LOOP INVARIANTS. CONSIDER A SEGMENT

```
WHILE CONDITION
      S
```

S IS REPEATEDLY EXECUTED UNTILL CONDITION BECOMES FALSE

p IS A LOOP INVARIANT IF $(p \wedge \text{CONDITION})\{S\}p$

RULE OF INFERENCE:

$$\frac{(p \wedge \text{CONDITION})\{S\}p}{p\{\text{WHILE CONDITION } S\}(\neg \text{CONDITION} \wedge p)}$$

EXAMPLE. VERIFY THAT THE SEGMENT

```
i := 1
factorial := 1
WHILE i<n
BEGIN
   i := i+1
   factorial := factorial * i
END
```

PRODUCES $factorial = n!$

SOLUTION. 1) CHECK THAT $p(i)$ IS A LOOP INVARIANT

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

INDUCTION ON i. $(p(i) \wedge \text{CONDITION}(i))\{S\}\,p(i+1) = A(i)$

$i=1$

$$A(1) \Leftarrow \begin{cases} p(1) \wedge \text{condition}(1): & factorial = 1 \wedge \cancel{1 \leq n} \wedge 1 < n \\ p(2): & factorial = 2 \wedge 2 \leq n \end{cases}$$

$i=k$

$$A(k) \Rightarrow A(k+1) \begin{cases} factorial = k! \wedge \cancel{k \leq n} \wedge k < n & p(k) \wedge \text{condition} \\ factorial = (k+1)! \wedge k \leq n & p(k+1) \end{cases}$$

2) CONCLUDE: $p\{\text{WHILE } i<n \; S\}[i \geq n \wedge p]$

$\Downarrow$ $i=n$ $\quad$ $\Downarrow$ $p(n) \implies factorial = n!$

HW 3.5: 2, 4, 10