BITS Pilani
Pilani Campus

MODULE: **PROGRAM VERIFICATION**

Floyd-Hoare Logic: Meta-Rule and Examples

Floyd-Hoare Logic

- Meta-Rule:

$$|-_\Delta \varphi' \text{ --> } \varphi \qquad <\varphi, \textbf{S}, \psi> \qquad |-_\Delta \psi \text{ --> } \psi'$$

$$<\varphi', \textbf{S}, \psi'>$$

- Alternatively,

  /* φ' */

  /* *Prove* φ from φ' */

  S

  /* ψ */

  /* *Prove* ψ' from ψ */

- This rule allows for *logical inferences between statements* in the program.
- The **proof system (Δ)** would be:
  - any proof system (such as Natural Deduction) for *predicate logic* with
  - added rules for the *domain of computation* e.g. **integers**

Floyd-Hoare Logic: Examples

- Example C2:
  - Re-do Example C1 so that m is the minimum of x and y

Floyd-Hoare Logic: Examples

Example C3:

```
/*  Pre: ? */
if (x % 2 == 0)
then {  y = y + 2; }
else { y = y + 1; }
/* Post:  (y > x) ∧ (y % 2 = 0)  */
```

Floyd-Hoare Logic: Examples

Ex C3:

```
/*  Pre:  ? */
if (x % 2 == 0)
then {  y = y + 2; }
else { y = y + 1; }
/* Post:
```
$(y > x) \wedge (y \% 2 = 0)$ */

- <u>then-case:</u>

$$/* (y+2 > x) \wedge (y+2)\%2=0$$
$$\text{i.e.} \ (y+2 > x) \wedge y\%2=0$$
*/

y = y + 2

/* $(y > x) \wedge (y \% 2 = 0)$ */

Floyd-Hoare Logic: Examples

Ex C3:

```
/*  Pre:  ? */
if (x % 2 == 0)
then {  y = y + 2; }
else { y = y + 1; }
/* Post:
```

$(y > x) \wedge (y \% 2 = 0)$ */

- then-case:

    /* $(y+2 > x) \wedge (y+2)\%2=0$

        i.e. $(y+2>x) \wedge y\%2=0$

    */

    y = y + 2

    /* $(y > x) \wedge (y \% 2 = 0)$ */

- else-case:

    /* $(y+1 > x) \wedge (y+1)\%2=0$

        i.e. $(y+1>x) \wedge \neg(y\%2=0)$

    */

    y = y + 2

    /* $(y > x) \wedge (y \% 2 = 0)$ */

Floyd-Hoare Logic: Examples

Ex C3:

```
/* Pre: ? */
if (x % 2 == 0)
then {  y = y + 2; }
else { y = y + 1; }
/* Post:
(y > x) ∧ (y % 2 = 0)  */
```

if-statement

Given post-condition:

$(y > x) \wedge (y \% 2 = 0)$

the precondition would be **φ**

i.e. $(y + 1 > x) \wedge (x\%2 = y\%2)$

• then-case:

/* $(y+2 > x) \wedge (y+2)\%2 = 0$
 i.e. $(y+2 > x) \wedge y\%2 = 0$
  <-- $\underbrace{(y+1 > x)}_{φ} \wedge \underbrace{y\%2 = 0}_{B[y/x]}$
*/

y = y + 2

/* $(y > x) \wedge (y \% 2 = 0)$ */

• else-case:

/* $(y+1 > x) \wedge (y+1)\%2 = 0$
 i.e. $\underbrace{(y+1 > x)}_{φ} \wedge \underbrace{\neg(y\%2 = 0)}_{\neg B[y/x]}$
*/

y = y + 1

/* $(y > x) \wedge (y\%2 = 0)$ */

Exercise: Initialize the variable y so as to satisfy the pre-condition (assuming x is the input).

Floyd-Hoare Logic: Examples

Exercise C3a:

/* Pre: ? */

if (x % 2 == 0)

then { y = x + 2; }

else { y = x + 1; }

/* Post:

(y > x) $\wedge$ (y % 2 = 0) */

1. Derive the precondition in this modified version of Exercise C3.
2. Do you require an _initializer_ for y?
   - If so, what is it?
   - If not, why not?