A Core programming Language & Hoare triples

- For sake of developing program-proving concepts, we define a core prog. lang.
- It consists of: Integer expressions, Boolean expressions, and commands.

Integer expressions $E = n \mid x \mid -E \mid E_1+E_2 \mid E_1-E_2 \mid E_1 * E_2$

Boolean expressions, $B = T \mid F \mid \neg B \mid B_1 \wedge B_2 \mid B_1 \vee B_2 \mid E_1 < E_2$

Commands, $S = x{:=}\underset{\substack{\uparrow \\ \text{integer exp.}}}{E} \mid S_1 ; S_2 \mid \text{if } \underset{\substack{\uparrow \\ \text{Boolean exp.}}}{B} \text{ then } S_1 \text{ else } S_2 \mid \text{while } B \text{ do } S.$

Example:

$$P \quad \left\{ \begin{array}{l} y:=1; \\ z:=0; \\ \text{while } (z \neq x) \text{ do } \{ \\ \ z:=z+1; \\ \ y:=y*z \,\} \end{array} \right.$$

Hoare triple:

$$\underset{\substack{\uparrow \\ \text{pre-condition}}}{[x>0]} \quad P \quad \underbrace{[y = x!]}_{\text{post-condition}}$$

Proving correctness requires establishing P transforms "pre-condition" to "post-condition".

$$(E_1 = E_2) \Leftrightarrow \neg(E_1 < E_2) \wedge \neg(E_2 < E_1)$$