A Floyd-Hoare Logic

Simple Imperative Programming Language:

$$C ::= I\text{:=}E \mid C \text{ ; } C \mid \text{if } B \text{ then } C \text{ else } C \text{ fi} \mid \text{while } B \text{ od } C \text{ od } C$$

where I ranges over program identifiers, E ranges over program arithmetic expressions, and B ranges boolean-valued expressions.

Rules:

Assignment Axiom:

$$\frac{}{\{P[e/x]\}; x\text{:=}e\{P\}}$$

Sequencing Rule:

$$\frac{\{P\}C_1\{Q\} \quad \{Q\}C_2\{R\}}{\{P\}C_1 \text{ ; } C_2\{R\}}$$

If_then_else Rule:

$$\frac{\{P \text{ and } B\}C_1\{Q\} \quad \{P \text{ and (not } B)\}C_2\{Q\}}{\{P\}\text{if } B \text{ then } C_1 \text{ else } C_2 \text{ fi}\{Q\}}$$

While Rule:

$$\frac{\{P \text{ and } B\}C\{Q\}}{\{P\}\text{while } B \text{ do } C \text{ od}\{P \text{ and (not } B)\}}$$

Precondition Strengthening:

$$\frac{P \implies P' \quad \{P'\}C\{Q\}}{\{P\}C\{Q\}}$$

Postcondition Weakening

$$\frac{\{P\}C\{Q'\} \quad Q' \implies Q}{\{P\}C\{Q\}}$$