Control Structures and Proof Rules

We will consider the three basic control structures of imperative programming. A command $C$ denotes a (possibly compound) statement formed from the atomic statements using the control structures. A series of commands can be enclosed in `begin ... end` to form a single syntactic unit. The control structures we investigate are (where $B$ is a Boolean-valued expression):

- sequential execution — $C_1; C_2$
- conditional — `if $B$ then $C_1$ else $C_2$`
- while-loop — `while $B$ do $C$`

Each of these control structures has an associated proof rule — based on properties of its components, we infer properties of the compound statement. These are as follows:

**Sequential rule**

$$\frac{|- {\langle P \rangle \ C_1 \ \{Q\}}, \ |- \ {\langle Q \rangle \ C_2 \ \{R\}}}{|- {\langle P \rangle \ C_1 ; C_2 \ \{R\}}}$$

**Conditional rule**

$$\frac{|- {\langle P \parallel B \rangle \ C_1 \ \{R\}}, \ |- \ {\langle P \parallel \parallel B \rangle \ C_2 \ \{R\}}}{|- {\langle P \rangle \ \text{if} \ B \ \text{then} \ C_1 \ \text{else} \ C_2 \ \{R\}}}$$

**While rule**

$$\frac{|- {\langle P \parallel B \rangle \ C \ \{P\}}}{|- {\langle P \rangle \ \text{while} \ B \ \text{do} \ C \ \{P \parallel \parallel B\}}}$$