CSC 236: Program Correctness Axioms

Where A, A0, A1 are computer programs, P, Q, R are propositions, and b is a boolean expression in the program variables,

**Base rules (programs)**
*(where “skip” is the null program)*

\[
\{P\} \text{skip}\{Q\} \iff P \rightarrow Q
\]

\[(\text{skip};A) = (A;\text{skip}) = A\]

**Base rule (specifications)**
You can conclude that \{false\}A\{P\} (for any A and P).

**Assignment rule**
If you know that \(P \rightarrow (Q[x:=E])\),
you can conclude that \{P\}x:=E\{Q\}.

**Sequential composition rule**
If you know that \{P\}A0\{Q\} and \{Q\}A1\{R\},
you can conclude that \{P\}A0;A1\{R\}.

**Conditional rules**
If you know that \(\{P \land b\}A\{Q\} \land P \land \lnot b \rightarrow Q\),
you can conclude that \{P\} if b then A end if \{Q\}.
If you know that \(\{P \land b\}A0\{Q\} \land \{P \land \lnot b\}A1\{Q\},\)
you can conclude that \{P\} if b then A0 else A1 end if \{Q\}.

**Loop rule**
If you know that \(\{P \land b\}A\{P\},\)
you can conclude that \{P\} while b do A end while \{P \land \lnot b\}.

**Note** that we do not use induction to prove partial correctness of a loop in this scheme. The loop inference rule in effect contains the induction within itself. However, we still need to prove termination separately (probably with induction), unless we use the “loop rule with variant”:

**Loop rule with variant**
Where A is a computer program, P is a proposition, e is an integer expression in the program variables, and “prev” is a constant value for each loop iteration,
If you know that \(\{P \land (e>0) \land (prev=e)\} \land \{P \land (e < prev)\},\)
you can conclude that \{P\} while e>0 do A end while \{P \land (e \leq 0)\}.