

## The Floyd-Hoare Method

If you were taught how to prove partial correctness of programs, you probably learned the Floyd-Hoare method. A Floyd-Hoare proof is equivalent to an inductive-invariance proof such as the one for algorithm *Euclid*. It's easy to convert either kind of proof to the other.

The Floyd-Hoare method proves that if a program begins in a state satisfying a precondition  $P$  and it terminates, then it does so in a state satisfying a postcondition  $Q$ . This is proved in  $\text{TLA}^+$  by proving the invariance of  $(pc = \text{“Done”}) \Rightarrow Q$  for an algorithm whose initial predicate is  $\text{Init} \wedge P$ , where  $\text{Init}$  is the initial predicate of the algorithm's  $\text{TLA}^+$  translation.

To write a Floyd-Hoare proof of a PlusCal program, we would annotate each labeled statement with a state predicate  $P_c$ , where  $c$  is the statement's label. We would also put a predicate  $P_{\text{Done}}$  at the end of the algorithm. The Floyd-Hoare proof for this annotation would be equivalent to an inductive-invariance proof, where the inductive invariant is the conjunction of the formulas

$$(pc = \text{“c”}) \Rightarrow P_c$$

for all annotations  $P_c$ .