Example Axiomatic Proof

The Fibonacci numbers are a sequence of integers defined recursively by
\[ \text{fib}(1) = \text{fib}(2) = 1, \text{ and } \]
\[ \text{fib}(N) = \text{fib}(N-1) + \text{fib}(N-2), \text{ for } N>2. \]
When defined in this usual way, the naturally corresponding recursive program is clearly correct but highly inefficient. We prove that the following iterative program fragment is correct — it’s totally correct, but we only exhibit the partial correctness proof.

\[
\begin{align*}
\{ N\geq 1 \} \\
\text{NEW} := 1; \quad \text{OLD} := 1; \quad I := 2; \\
\{ \mathcal{P} \} \\
\textbf{while} \ I < N \textbf{ do} \\
\quad \textbf{begin} \ I := I + 1; \quad \text{NEW} := \text{NEW} + \text{OLD}; \\
\quad \quad \text{OLD} := \text{NEW} - \text{OLD} \quad \textbf{end} \\
\{ \text{NEW} = \text{fib}(N) \}
\end{align*}
\]

Proof:

Step 1: discover the loop invariant \( \mathcal{P} \)

Take \( \mathcal{P} \equiv (2 \leq I \leq N \land \text{NEW} = \text{fib}(I) \land \text{OLD} = \text{fib}(I-1)) \lor (I = 2 \land N = 1 \land \text{NEW} = 1) \)

Step 2: Show \( \vdash \{ N \geq 1 \} \text{NEW} := 1; \quad \text{OLD} := 1; \quad I := 2 \quad \{ \mathcal{P} \} \)

Try these details as an exercise - just uses Assign and SEQ.

Step 3: Show \( \vdash \{ \mathcal{P} \} \textbf{ while } ... \quad \{ \text{NEW} = \text{fib}(N) \} \)

This step is established through several intermediate steps.

Step 3A: Find \( \mathcal{Q}_1 \) and \( \mathcal{Q}_2 \) to show

\( \vdash \{ \mathcal{P} \land I < N \} \)

\[
\begin{align*}
\textbf{begin} \ I := I + 1; \\
\quad \{ \mathcal{Q}_1 \} \text{NEW} := \text{NEW} + \text{OLD}; \\
\quad \{ \mathcal{Q}_2 \} \text{OLD} := \text{NEW} - \text{OLD} \quad \textbf{end} \\
\quad \{ \mathcal{P} \}
\end{align*}
\]

Step 3Ai: formulate \( \mathcal{Q}_1 \)
Take $Q_1 \equiv 3 \leq I \leq N \land NEW = \text{fib}(I-1) \land OLD = \text{fib}(I-2)$

**Step 3Aii:** Show $\vdash \{P \land I < N\} I := I+1 \{Q_1\}$

It can be seen that $(P \land I < N) \Rightarrow Q_1[I \rightarrow I+1]$ so by Assign axiom and rule for and Strengthening pre-conditions, step 3Aii holds.

**Step 3Aiii:** formulate $Q_2$

Take $Q_2 = 3 \leq I \leq N \land NEW = \text{fib}(I) \land OLD = \text{fib}(I-2)$

**Step 3Aiv:** show $\vdash \{Q_1\} NEW := NEW + OLD \{Q_2\}$

Try this - direct application of Assign axiom.

**Step 3Av:** show $\vdash \{Q_2\} OLD := NEW - OLD \{P\}$

One can see that $Q_2 \Rightarrow P[OLD \rightarrow NEW - OLD]$ so that by Assign axiom and rule for Strengthening pre-conditions, this step is proven

**Step 3Avi:** by steps 3Aii, 3Aiv, and 3Av and the rule for sequential execution (applied twice), the proof of step 3A is complete.

**Step 3B:** by step 3A and the rule for while loops we have $\vdash \{P\} \text{while } I < N \text{ do begin } \ldots \text{ end } \{P \land I \geq N\}$. Now, $P \land I \geq N$ implies either

$I = 2 \land N = 1 \land NEW = 1$, and hence $NEW = \text{fib}(N)$

or

$2 \leq I = N \land NEW = \text{fib}(I) \land OLD = \text{fib}(I-1)$, and hence $NEW = \text{fib}(N)$

**Step 4:** By steps 2 and 3 and the post-condition Weakening rule, the program is proven.