Example Axiomatic Program Proof

The Fibonacci numbers are a sequence of integers defined recursively by

fib(0) = 0, fib(1) = 1, and

fib(N) = fib(N-1)+fib(N-2), for N>1.

The naturally corresponding recursive program (in any language) for this definition is clearly correct but so *highly* inefficient that it is of no practical use, even for small (e.g., two digit) arguments. We prove that the following iterative program fragment in Louden's Sample language is correct (its performance is clearly directly proportional to the size of the argument N).

```
        { N≥0 }
NEW:= 1; OLD:= 0; I:= 0;
        { ℙ }
    while N-I do
    I:= I+1; NEW:= NEW+OLD;
    OLD:= NEW-OLD od
        { OLD = fib(N) }
```

Proof (read ⊢ as "it is provable that")

Step 0: discover the loop invariant ℙ

Informally the idea of the loop is that as I is incremented, the variables NEW and OLD are revised to maintain the value of fib(I) and fib(I+!). We also include a technical condition relating I and N that's needed in the last step.

Take ℙ ≡ (0≤I≤N ∧ NEW=fib(I+1) ∧ OLD=fib(I))

Step 1: Show ⊢ { N≥0 } NEW:=1; OLD:=0; I:=0 { ℙ }

Exercise — takes several steps using ASN and SEQ.

Step 2: Show ⊢ { ℙ } **while** ... { OLD=fib(N) } (i.e., prove the post-condition)

This step is established through several intermediate steps.

Step 2A: Find $\mathbb{Q}_1$ and $\mathbb{Q}_2$ to show (i.e., ℙ is a loop invariant)

```
⊢ { ℙ ∧ N-I>0 }
 I:=I+1;
{ ℚ1 } NEW:= NEW+OLD;
{ ℚ2 } OLD:= NEW-OLD
        { ℙ }
```

Example Axiomatic Program Proof

Step 2Ai: formulate $\mathbb{Q}_1$

After I is incremented, but NEW and OLD have not yet been changed, the Fibonacci indicies of NEW and OLD are one step behind.

Take $\mathbb{Q}_1 \equiv 0 \le I \le N \wedge NEW=fib(I) \wedge OLD=fib(I-1)$

Step 2Aii: Show $\vdash \{ \mathbb{P} \wedge I< N \}\ I:= I+1\ \{ \mathbb{Q}_1 \}$

It can be seen that $(\mathbb{P} \wedge I<N) \rightarrow \mathbb{Q}_1[I \leftarrow I+1]$ so by ASN and STR, step 2Aii holds.

Step 2Aiii: formulate $\mathbb{Q}_2$

At this point, the index I and the variable NEW have been updated, but the variable OLD is still a step behind.

Take $\mathbb{Q}_2 \equiv 0 \le I \le N \wedge NEW=fib(I+1) \wedge OLD=fib(I-1)$

Step 2Aiv: show $\vdash \{ \mathbb{Q}_1 \}\ NEW:= NEW+OLD\ \{ \mathbb{Q}_2 \}$

This is a direct application ofASN.

Step 2Av: show $\vdash \{ \mathbb{Q}_2 \}\ OLD:= NEW-OLD\ \{ \mathbb{P} \}$

One can see that $\mathbb{Q}_2 \rightarrow \mathbb{P}[OLD \leftarrow NEW-OLD]$ so that by ASN and STR, this step is proven

Step 2Avi: by steps 2Aii, 2Aiv, and 2Av and SEQ (applied twice), the proof of step 2A is complete.

Step 2B: by step 2A and WHL we have

$\vdash \{ \mathbb{P} \}$ **while** ... $\{ \mathbb{P} \wedge N-I \le 0 \}$. Now, $\mathbb{P} \wedge N-I \le 0$ implies (this is where we need $0 \le I \le N$ included in the loop invariant)

$I=N \wedge OLD=fib(I)$/

Therefore, $\vdash \mathbb{P} \wedge I \ge N \wedge OLD=fib(N)$ (i.e., the value of I is immaterial at the end).

Step 3: By steps 1 and 2 and WKN , the program is proven.

This presentation has illustrated how to *discover* the program proof and determine the needed steps. A valid logic proof would require reordering all the individual steps so that each is either an axiom or is derived from *previous* steps by a rule of inference.