Example program proof — while rule

In this example, we prove a program assertion establishing partial correctness of a program fragment computing the factorial of an integer.

```
        { N ≥ 0 }
M:= 0;  F = 1;
while M < N do
begin  M := M+1;   F := M*F  end
        { F = N! }
```

To construct the proof of this program, we need to determine an intermediate formula ℙ that will serve as a loop invariant. This will describe the relationship of the variable values at the intermediate point noted below.

```
        { N ≥ 0 }
M:= 0;  F = 1;
          ℙ
while M < N do
begin  M := M+1;   F := M*F  end
        { F = N! }
```

We take ℙ to be the formula { $F = M! \wedge 0 \le M \le N$ }. Then the proof is as follows:

1. By two applications of the axiom of assignment and the sequential execution rule
   | { $1 = 0! \wedge 0 \le N$ } M:= 0; F = 1 ℙ
   and the pre-condition is logically equivalent to the pre-condition of the program.
2. By two applications of the axiom of assignment and the sequential execution rule
   | { $(M+1)*F = (M+1)! \wedge 0 \le M+1 \le N$ } M := M+1; F := M*F ℙ
3. Since ℙ $\wedge M < N \Rightarrow (M+1)*F = (M+1)! \wedge 0 \le M+1 \le N$, by the rule for strengthening the pre-condition, ℙ is an invariant for the while loop.

4. By step 3 and the while rule

   $\mathbb{P}$

   while M < N **do**
   begin M := M+1; F := M*F **end**

   $\mathbb{P} \wedge M \geq N$

5. Since $\mathbb{P} \wedge M \geq N \Rightarrow F = N!$, by the rule for weakening the post-condition, the program proof is complete.