Proof Rules and Proofs for Correctness Triples

Part 1: Axioms, Sequencing, and Auxiliary Rules

CS 536: Science of Programming, Fall 2022

2022-11-07: p.4

A. Why

- We can't generally prove that correctness triples are valid using truth tables.
- We need proof axioms for atomic statements (***skip*** and assignment) and inference rules for compound statements like sequencing.
- In addition, we have inference rules that let us manipulate preconditions and postconditions.

B. Objectives

At the end of this practice activity you should

- Be able to match a statement and its conditions to its proof rule.

C. Problems

Use the vertical format to display rule instances. Below, ^ means exponentiation.

1. Consider the triples $\{p_1\}\ x := x+x\ \{p_2\}$ and $\{p_2\}\ k := k+1\ \{x = 2\text{^}k\}$ where p_1 and p_2 are unknown.
    a. Find values for p_1 and p_2 that make the triples provable. (Hint: Use *wp*.)
    b. What do you get if you combine the triples using the sequence rule? Show the complete three-line proof. (Include the rules for the two assignments before using sequence.)
    c. Add (two more) lines to the proof to strengthen the precondition to be $x = 2\text{^}k$ instead of p_1.
    d. Rewrite the proof so that instead of forming the sequence and then strengthening its precondition to $x = 2\text{^}k$, we strengthen the precondition of $x := x+x$ to be $x = 2\text{^}k$ before combining with $k := k+1$ to form the sequence.
    e. Write a new proof that uses *sp* on the two assignments (instead of *wp*), then forms the sequence and then weakens the postcondition.
    f. Write a new proof that again uses *sp* but this time simplify the postcondition of each assignment (using weakening) before forming the sequence.

2. (Establishing $x = 2^k$)
   a. Write a proof of $\{T\}\ x := 1;\ k := e\ \{x = 2^k\}$ that uses *wp* to calculate p and q for $\{p\}\ k := e\ \{x = 2^k\}$ and $\{q\}\ x := 1\ \{p\}$, forms the sequence, and strengthens the initial precondition to T. Also, what value should we use for e?
   b. Repeat, but on the sequence $\{T\}\ k := e;\ x := 1; \{x = 2^k\}$. (No change to e is needed.)
   c. Now give a proof for $\{T\}\ k := 1;\ x := e\ \{x = 2^k\}$ that uses *sp* on each assignment and weakens the final postcondition to $x = 2^k$. What value do you want for e?
   d. One more variation: Use *sp* on $k := 1$ and *wp* on $x := \ldots$.

3. The proof below is incomplete.

1.	$\{p\}\ S_1\ \{q\}$	assumption 1
2.	$q \rightarrow q'$	assumption 2
3.	???	???
4.	$\{q'\}\ S_2\ \{r\}$	assumption 3
5.	$\{p\}\ S_1;\ S_2\ \{r\}$	???

   a. Fill in the missing parts to get a complete proof.
   b. Turn the proof into a derived proof rule by changing "assumption" to "antecedent", dropping line 3, and using "extended sequence 1, 2, 3" for the last line. What is your result?

Solution to Practice 14 (Proof Rules and Proofs, pt.1)

1. (Preconditions for $x = 2\textasciicircum k$ postcondition)

   a. $p_2 \equiv wp(k := k+1,\ x = 2\textasciicircum k) \equiv x = 2\textasciicircum(k+1)$.
   $p_1 \equiv wp(x := x+x,\ p_2) \equiv wp(x := x+x,\ x = 2\textasciicircum(k+1)) \equiv x+x = 2\textasciicircum(k+1)$.

   b. The full proof is:

1.	$\{x = 2\textasciicircum(k+1)\}\ k := k+1\ \{x = 2\textasciicircum k\}$	assignment (backward)
2.	$\{x+x = 2\textasciicircum(k+1)\}\ x := x+x\ \{x = 2\textasciicircum(k+1)\}$	assignment (backward)
3.	$\{x+x = 2\textasciicircum(k+1)\}\ x := x+x;\ k := k+1\ \{x = 2\textasciicircum k\}$	sequence 2, 1

   c. To make the precondition $x = 2\textasciicircum k$, we have to strengthen the precondition of line 3. We need two more lines of proof.

   (1 - 3 same as in part b)

4.	$x= 2\textasciicircum k \rightarrow x+x = 2\textasciicircum(k+1)$	predicate logic
5.	$\{x = 2\textasciicircum k\}\ x := x+x;\ k := k+1\ \{x = 2\textasciicircum k\}$	precond. strength. 4, 3

   d. We need to reorder the proof lines to strengthen the precondition of $x := x+x$ before combining it with $k := k+1$:

1.	$\{x = 2\textasciicircum(k+1)\}\ k := k+1\ \{x = 2\textasciicircum k\}$	assignment (backward)
2.	$\{x+x = 2\textasciicircum(k+1)\}\ x := x+x\ \{x = 2\textasciicircum(k+1)\}$	assignment (backward)
3.	$x= 2\textasciicircum k \rightarrow x+x = 2\textasciicircum(k+1)$	predicate logic
4.	$\{x = 2\textasciicircum k\}\ x := x+x\ \{x = 2\textasciicircum(k+1)\}$	precond. strength. 3, 2
5.	$\{x = 2\textasciicircum k\}\ x := x+x;\ k := k+1\ \{x = 2\textasciicircum k\}$	sequence 2, 1

   e. If we use *sp* on the assignments and weaken the postcondition of the sequence, we get:

1.	$\{x = 2\textasciicircum k\}\ x := x+x\ \{x_0 = 2\textasciicircum k \wedge x = x_0+x_0\}$	assignment (forward)
2.	$\{x_0 = 2\textasciicircum k \wedge x = x_0+x_0\}\ k := k+1\ \{q_0\}$ where $q_0 \equiv x_0 = 2\textasciicircum k_0 \wedge x = x_0+x_0 \wedge k = k_0+1$	assignment (forward)
3.	$\{x = 2\textasciicircum k\}\ x := x+x;\ k := k+1\ \{q_0\}$	sequence 2, 1
4.	$q_0 \rightarrow x = 2\textasciicircum k$	predicate logic
5.	$\{x = 2\textasciicircum k\}\ x := x+x;\ k := k+1\ \{x = 2\textasciicircum k\}$	postcond. weak. 3, 4

   f. If we use *sp* but weaken the postconditions as we go, we get:

1.	$\{x = 2\textasciicircum k\}\ x := x+x\ \{x_0 = 2\textasciicircum k \wedge x = x_0+x_0\}$	assignment (forward)
2.	$x_0 = 2\textasciicircum k \wedge x = x_0+x_0 \rightarrow x/2 = 2\textasciicircum k$	predicate logic
3.	$\{x = 2\textasciicircum k\}\ x := x+x\ \{x/2 = 2\textasciicircum k\}$	postcond. weak, 1, 2
4.	$\{x/2 = 2\textasciicircum k\}\ k := k+1\ \{x/2 = 2\textasciicircum k_0 \wedge k = k_0+1\}$	assignment (forward)
5.	$x/2 = 2\textasciicircum k_0 \wedge k = k_0+1 \rightarrow x = 2\textasciicircum k$	predicate logic
6.	$\{x/2 = 2\textasciicircum k\}\ k := k+1\ \{x = 2\textasciicircum k\}$	postcond. weak, 4, 5
7.	$\{x = 2\textasciicircum k\}\ x := x+x;\ k := k+1\ \{x = 2\textasciicircum k\}$	sequence 3, 6

2. (Proofs of *{T} x := 1*; *k := e {x = 2^k}*.)

   a. (Use *wp* twice, form the sequence, and strengthen the precondition to *T*.)

1.	*{x = 2^e} k := e {x = 2^k)*	assignment (backward)
2.	*{1 = 2^e} x := 1 {x = 2^e}*	assignment (backward)
3.	*{1 = 2^e} x := 1; k := e {x = 2^k)*	sequence 2, 1
	(Note we need *e = 0*)	
4.	*T* → *1 = 2^e*	predicate logic
5.	*{T} x := 1; k := e {x = 2^k)*	precond. strength. 4, 3

   b. (Prove *{T} k := e; x := 1 {x = 2^k}* in the same way, with no change to *e*.)

1.	*{1 = 2^k} x := 1 {x = 2^k)*	assignment (backward)
2.	*{1 = 2^0} k := 0 {1 = 2^k}*	assignment (backward)
	(Again, *e = 0*)	
3.	*{1 = 2^0} k := 0; x := 1 {x = 2^k)*	sequence 2, 1
4.	*T* → *1 = 2^0*	predicate logic
5.	*{T} k := 0; x := e {x = 2^k)*	precond. strength. 4, 3

   c. (Prove *{T} k := 1*; *x := e {x = 2^k}* using *sp* and ending with postcondition weakening.)

1.	*{T} k := 1 {k = 1}*	assignment (forward)
2.	*{k = 1} x := e {k = 1 ∧ x = e}*	assignment (forward)
3.	*k = 1 ∧ x = e* → *x = 2^k*	predicate logic
4.	*{k = 1} x := e {k = 1 ∧ x = e}*	postcond. weak. 2, 3 [2022-11-07]
5.	*{T} k := 1; x := e {x = 2^k}*	sequence 1, 4

   This time, *e = 2*, since we need *x = 2^k* with *k = 1.*

   d. (Prove *{T} k := 1*; *x := e {x = 2^k}* using *sp* on first assignment, *wp* on second.)

1.	*{T} k := 1 {k = 1}*	assignment (forward)
2.	*{e = 2^k} x := e {x = 2^k}*	assignment (backward)
3.	*k = 1* → *e = 2^k*	predicate logic
4.	*{k=1} x := e {x = 2^k} [2022-11-07]*	precond. strength. 3, 2
5.	*{T} k := 1*; *x := e {x = 2^k}*	sequence 1, 4

3. (Derive an extended sequence rule)

   a. Filling in the missing parts gives

1.	*{p}* S_1 *{q}*	antecedent 1
2.	*q* → *q′*	antecedent 2
3.	*{p}* S_1 *{q′}*	postcond. weak. 1, 3
4.	*{q′}* S_2 *{r}*	antecedent 3
5.	*{p}* S_1; S_2 *{r}*	sequence 4, 3

b. After we change "assumption" to "antecedent", change the last line's reason to "extended sequence" and drop the remaining line(s), we get a derived rule:

1.	$\{p\}\ S_1\ \{q\}$	antecedent 1
2.	$q \to q'$	antecedent 2
3.	$\{q'\}\ S_2\ \{r\}$	antecedent 3
4.	$\{p\}\ S_1;\ S_2\ \{r\}$	extended sequence 1, 2, 3