Solution to Homework 5 — Strongest Postconditions, Proof Rules

To help remind that you doesn't matter what logical constant you use to name the old value of a variable, I'll use different ones below: maybe `x'`, maybe x_0, and maybe something else.

Lecture 13: Strongest Postconditions

1. (Validity under partial correctness but not total correctness)

   For {T} *S* {*sp*(*p*, *S*)} to be valid for total correctness, it has to be valid for partial correctness and also always terminate. So to be invalid for total correctness, we a state in which either {T} *S* {*sp*(*p*, *S*)} is not partially correct or *S* does not terminate. But we're given that {T} *S* {*sp*(*p*, *S*)} is partially correct, so we need a state in which *S* doesn't terminate (gets a runtime error or diverges). A couple of examples:

   {T} x := 0**;** y := 2 / x {x = 0 ∧ y = 2/x}

   {T} x := −1**; while** x ≠ 0 **do** x **:=** x−1 **od**; x := 3 {x = 3}

   (This last one is a bit sneaky; we don't know how to calculate the *sp* of a loop, but *sp* of the sequence above has to be *sp*(x := 3).)

   (The relationship with ⊨ {T} *S* {F} is that σ ⊨ {T} *S* {F} means *S* doesn't terminate when run in σ, since if it terminated, it would have to be in a state in which false is true.)

2. (Calculate *sp*, no logical simplification)

   sp(i < j ∧ j−i ≤ n, i **:=** f(i+j); j := g(i*j))

   ≡ *sp*(*sp*(i < j ∧ j-i < n, i **:=** f(i+j)), j **:=** g(i*j))

   ≡ *sp*(i_0 < j ∧ j-i_0 ≤ n ∧ i = f(i_0+j), j **:=** g(i*j))

   ≡ i_0 < j_0 ∧ j_0−i_0 ≤ n ∧ ∧ i = f(i_0+j_0) ∧ j := g(i*j_0)

3. (Calculate *sp*, logical simplifications allowed where specified)

   3a. *sp*(x = 2^k, x **:=** x/2) ≡ x′ = 2^k ∧ x = x′/2 ⇒ x = 2^(k−1) [note x′ was dropped]

   wp(x **:=** x/2, x = 2^k) ≡ x/2 = 2^k ⇔ x = 2^(k+1)

   3b. (*S* ≡ **if** even(x) **then** x **:=** x+1 **fi**)

   sp(x = x_0, **if** even(x) **then** x **:=** x+1 **else skip fi**)

   ≡ *sp*(x = x_0 ∧ even(x), x **:=** x+1) ∨ *sp*(x = x_0 ∧ odd(x), **skip**)

   ≡ (even(x_0) ∧ x = x_0+1) ∨ (x = x_0 ∧ odd(x))

   If you don't mind losing the relationship with x_0, this last predicate implies odd(x).

   wp(*S*, odd(x))

   ≡ *wp*(**if** even(x) **then** x **:=** x+1 **else skip fi**, odd(x))

   ≡ (even(x) → *wp*(x **:=** x+1, odd(x))) ∧ (odd(x) → *wp*(**skip**, odd(x)))

$\equiv (\text{even}(\mathtt{x}) \to \text{odd}(\mathtt{x+1})) \land (\text{odd}(\mathtt{x}) \to \text{odd}(\mathtt{x}))$

$\Leftrightarrow \text{odd}(\mathtt{x})$

3c. $(p \equiv \mathtt{L} < \mathtt{R} \land \mathtt{b[L]} \le \mathtt{x} < \mathtt{b[R]}$

and $S \equiv$ **if** $\mathtt{x} < \mathtt{b[M]}$ **then** $\mathtt{R} := \mathtt{M}$ **else** $\mathtt{L} := \mathtt{M}$ **fi**)

$sp(\mathtt{R} = \mathtt{R}_0 \land \mathtt{L} = \mathtt{L}_0 \land p, S)$

$\equiv sp(\mathtt{R} = \mathtt{R}_0 \land \mathtt{L} = \mathtt{L}_0 \land p \land \mathtt{x} < \mathtt{b[M]}, \mathtt{R} := \mathtt{M})$

$\lor\, sp(\mathtt{R} = \mathtt{R}_0 \land \mathtt{L} = \mathtt{L}_0 \land p \land \mathtt{x} \ge \mathtt{b[M]}, \mathtt{L} := \mathtt{M})$

$\equiv (\mathtt{L} = \mathtt{L}_0 \land \mathtt{L} < \mathtt{R}_0 \land \mathtt{b[L]} \le \mathtt{x} < \mathtt{b}[\mathtt{R}_0] \land \mathtt{x} < \mathtt{b[M]} \land \mathtt{R} = \mathtt{M})$

$\lor\, (\mathtt{R} = \mathtt{R}_0 \land \mathtt{L}_0 < \mathtt{R} \land \mathtt{b}[\mathtt{L}_0] \le \mathtt{x} < \mathtt{b[R]} \land \mathtt{x} < \mathtt{b[M]} \land \mathtt{L} = \mathtt{M})$

You didn't have to simplify, but if you wanted to, one possibility is

$\mathtt{L}_0 < \mathtt{R}_0 \land \mathtt{b}[\mathtt{L}_0] \le \mathtt{x} < \mathtt{b}[\mathtt{R}_0]$

$\land\, (\mathtt{x} < \mathtt{b[M]} \to \mathtt{L} = \mathtt{L}_0 \land \mathtt{R} = \mathtt{M}) \land (\mathtt{x} < \mathtt{b[M]} \to \mathtt{R} = \mathtt{R}_0 \land \mathtt{L} = \mathtt{M})$

$wp(S, p)$

$\equiv (\mathtt{x} < \mathtt{b[M]} \to wp(\mathtt{R} := \mathtt{M}, p)) \land (\mathtt{x} \ge \mathtt{b[M]} \to wp(\mathtt{L} := \mathtt{M}, p))$

$\equiv (\mathtt{x} < \mathtt{b[M]} \to p[\mathtt{M/R}])) \land (\mathtt{x} \ge \mathtt{b[M]} \to p[\mathtt{M/L}]))$

$\equiv (\mathtt{x} < \mathtt{b[M]} \to \mathtt{L} < \mathtt{M} \land \mathtt{b[L]} \le \mathtt{x} < \mathtt{b[M]}) \land (\mathtt{x} \ge \mathtt{b[M]} \to \mathtt{M} < \mathtt{R} \land \mathtt{b[M]} \le \mathtt{x} < \mathtt{b[R]})$

Lectures 14-15: Proof Rules

(Find predicates)

4. $p_1 \equiv \mathtt{x = 2\hat{}(k+1)} \land \mathtt{k+1} \le \mathtt{n}$, $p_2 \equiv \mathtt{2*x = 2\hat{}(k+1)} \land \mathtt{k+1} \le \mathtt{n}$, and

$p_3 \equiv p \land \mathtt{k} \ge \mathtt{n} \equiv \mathtt{x = 2\hat{}k} \land \mathtt{k} \le \mathtt{n} \land \mathtt{k} \ge \mathtt{n}$

1. $\{p_1\}$ `k := k+1` $\{p\}$ — assignment

   where $p \equiv \mathtt{x = 2\hat{}k} \land \mathtt{k} \le \mathtt{n}$ and $S \equiv$ `x := x*2; k := k+1`
2. $\{p_2\}$ `x := x*2` $\{p_1\}$ — assignment
3. $\{p_2\}$ `x := x*2; k := k+1` $\{p\}$ — sequence 2, 1
4. $p \land k < n \to p_2$ — pred logic
5. $\{p \land k < n\}$ `x := x*2; k := k+1` $\{p\}$ — pre str. 4, 3
6. $\{$**inv** $p\}$ **while** $\mathtt{k} < \mathtt{n}$ **do** S **od** $\{p_3\}$ — while, 3

5. $q_1 \equiv (\mathtt{r = X*Y-(x/2)*(2*y)})$ and $q_2 \equiv (\mathtt{r+y = X*Y-(x-1)*y})$ in

1. $\{q_1\}$ `x := x/2; y := 2*y` $\{\mathtt{r = X*Y-x*y}\}$ (*)
2. $\{q_2\}$ `x := x-1; r := r+y` $\{\mathtt{r = X*Y-x*y}\}$ (*)
3. $\{(\mathtt{r = X*Y-x*y} \land \text{even}(\mathtt{x}) \to q_1)$

   $\land\, (\mathtt{r = X*Y-x*y} \land \text{odd}(\mathtt{x}) \to q_2)\}$ — conditional 1, 2

   if even(x) **then** `x := x/2; r := 2*r`

   else `x := x-1; r := r+y` **fi** $\{\mathtt{X*Y = r-x*y}\}$

(*) Subproof used assignment, assignment, and sequence as in Question 4

6. $r_1 \equiv (\texttt{r} = \texttt{r}_0 \wedge \texttt{r} = \texttt{X*Y-x}_0\texttt{*y}_0 \wedge \texttt{even}(\texttt{x}_0) \wedge \texttt{x} = \texttt{x}_0\texttt{/2} \wedge \texttt{y} = \texttt{2*y}_0)$,
$r_2 \equiv (\texttt{y} = \texttt{y}_0 \wedge \texttt{r}_0 = \texttt{X*Y-x}_0\texttt{*y} \wedge \texttt{odd}(\texttt{x}_0) \wedge \texttt{x} = \texttt{x}_0\texttt{-1} \wedge \texttt{r} = \texttt{r}_0\texttt{+y})$, and $r_3 \equiv r_1 \vee r_2$ in

1. $\{\texttt{x} = \texttt{x}_0 \wedge \texttt{y} = \texttt{y}_0 \wedge \texttt{r} = \texttt{r}_0 \wedge \texttt{r} = \texttt{X*Y-x*y} \wedge \texttt{even(x)}\}$ (*)
   `x := x/2; y := 2*y` $\{r_1\}$
2. $\{\texttt{x} = \texttt{x}_0 \wedge \texttt{y} = \texttt{y}_0 \wedge \texttt{r} = \texttt{r}_0 \wedge \texttt{r} = \texttt{X*Y-x*y} \wedge \texttt{odd(x)}\}$ (*)
   `x := x-1; r := r+y` $\{r_2\}$
3. $\{\texttt{x} = \texttt{x}_0 \wedge \texttt{y} = \texttt{y}_0 \wedge \texttt{r} = \texttt{r}_0 \wedge \texttt{r} = \texttt{X*Y-x*y}\}$ conditional 1, 2
   if even(x) **then** `x := x/2; y := 2*y`
   else `x := x-1; r := r+y` **fi** $\{r_3\}$

(*) Subproof used assignment, assignment, and sequence