SE 504 (Formal Methods and Models)
Spring 2017
HW #4: Catenation, Selection, and Invariants
Due: 7:30pm, Monday March 6

Let S be a program and Q be a predicate (over the state space of S). The expression wp.S.Q (read "weakest precondition of S with respect to Q") refers to the weakest predicate P satisfying the Hoare triple {P} S {Q}. In other words

$$\{\text{P}\}\ \text{S}\ \{\text{Q}\}\ \equiv\ [P \Rightarrow \text{wp}.S.Q]$$

Among the laws pertaining to wp are these:

wp skip law: $[\text{wp}.\mathbf{skip}.Q \equiv\ Q]$

wp assignment law: $[\text{wp}.(x := E).Q \equiv\ Q(x := E)]$

wp catenation law: $[\text{wp}.(S_1; S_2).Q \equiv \text{wp}.S_1.(\text{wp}.S_2.Q)]$

The wp catenation law says, in effect, that the weakest solution to $\{?\}\ S_1; S_2\ \{Q\}$ is none other than wp.S_1.R (i.e., the weakest solution to $\{?\}\ S_1$ {R}), where R is wp.S_2.Q (i.e., the weakest solution to $\{?\}\ S_2\ \{Q\}$).

That is, to obtain the weakest precondition for the catenation $S_1; S_2$ (with respect to a postcondition Q), we first find the weakest precondition for S_2 (with respect to Q), which serves as our "intermediate assertion" between S_1 and S_2.

In problems 1-3, simplify the given expression as much as possible. Use the wp laws given above, as well as well-known theorems from arithmetic, algebra, and logic. Regarding Problem 2, note that catenation is associative, meaning that $(S_1; S_2); S_3$ and $S_1; (S_2; S_3)$ are equivalent programs. Problem 3, despite being worded differently, is the same kind of problem as the ones preceding it.

1. (10 points) $\text{wp}.(i := i - 2 * j;\ \ j := j + i).(2i \geq j)$

2. (10 points) $\text{wp}.(y := x - y;\ \ x := x - y;\ \ y := y + x).(x = Y\ \wedge\ y = X)$

3. (10 points) Determine the weakest predicate P that makes this Hoare Triple true:

$$\{P\}\ i := i - 1;\ \ sum := sum + b.i\ \{sum = (+j \mid i \leq j < \#b\ :\ b.j)\ \wedge\ 0 \leq i \leq \#b\}$$

4. (13 points) Calculate an expression E (containing no occurrences of rigid variable $\mathtt{C}$, of course) that makes the given Hoare Triple true.

$$\{\mathtt{C} = km + r\ \wedge\ m > 0\ \wedge\ \text{isOdd}.m\}\ r := E;\ \ k, m := 2 * k, (m - 1)\ \text{div}\ 2\ \{\mathtt{C} = km + r\}$$

You should find it necessary to make use of this theorem:

$$(m > 0\ \wedge\ \text{isOdd}.m) \Rightarrow (((m - 1)\ \text{div}\ 2) = (m - 1)/2)$$

Recall that, if **IF** is the program

$$\textbf{if } B_0 \rightarrow S_0 \ [] \ B_1 \rightarrow S_1 \textbf{ fi}$$

then $[\text{wp.IF}.Q \equiv (B_0 \vee B1) \wedge (B_0 \Rightarrow \text{wp}.S_0.Q) \wedge (B_1 \Rightarrow \text{wp}.S_1.Q)]$

Using the relationship between wp and Hoare Triples, from this it follows that
$\{P\}$ IF $\{Q\} \quad \equiv \quad [P \Rightarrow (B_0 \vee B_1)] \;\wedge\; \{P \wedge B_0\}\; S_0\; \{Q\} \;\wedge\; \{P \wedge B_1\}\; S_1\; \{Q\}$

5. (23 points) Prove this Hoare Triple:

$\{P \;\wedge\; i < \#b\}$
if $b.i \;>\; 0 \;\rightarrow\; sum \;:=\; sum + b.i;\; i := i+1$
$[] \; b.i \;\leq\; 0 \;\rightarrow\; i := i+1$
fi
$\{P \;\wedge\; i \leq \#b\}$

where $P : 0 \leq i \;\wedge\; sum \;=\; (+j \mid 0 \leq j < i \;\wedge\; b.j > 0 \;:\; b.j)$

6. (24 points)

Complete, and narrate, the development of a program that satisfies this Hoare Triple by "solving for" E, an unknown expression. Note that no single solution works in all cases, so it will be necessary to introduce a selection command. Explain your reasoning, using as a model the *Two Examples of Deriving Selection Commands* web page.

$\{P \;\wedge\; 1 \leq i < \#b\}\; k \;:= E;\; i \;:=\; i+1\; \{P \;\wedge\; 1 \leq i \leq \#b\}$

where $P :\; k \;=\; (\#j \mid 1 \leq j < i \;:\; b.(j-1) > b.j)$

Note that $(\#x \mid R \;:\; Q)$ is an abbreviation for $(\#x \mid R \wedge Q \;:\; 1)$.

7. (10 points) Suppose that you have a chocolate bar similar to the one shown in the image at www.gettyimages.com/detail/photo/chocolate-bar-with-path-royalty-free-image/157419404. That is, it has "squares" separated by little troughs so as to make it easy to break the bar into rectangular-shaped pieces. Assume that initially the bar is in one piece and has n squares. You are to keep splitting the bar until you are left with n pieces, each being one of the original squares. Each time you split a piece, you must split it entirely along one of its troughs, so as to obtain two pieces.

How many times will you perform a split before you end up with n pieces? Justify your answer by making an argument that is based upon an invariant property/relationship involving quantities that are relevant to the situation.