Assignment 1

Advanced Computing concepts for Engineering

Solution 2015.

Note that the work that you turn in for this assignment must represent your individual effort. You are welcome to help your fellow students to understand the material of the course and the meaning of the assignment questions, however, the answer that you submit must be created by you alone.

Q0[10] Given $\Sigma = \{\text{“}w\text{”} \mapsto \mathbb{R}, \text{“}x\text{”} \mapsto \mathbb{R}, \text{“}y\text{”} \mapsto \mathbb{R}, \text{“}z\text{”} \mapsto \mathbb{R}\}$, implement the following specification using a sequence of (nonparallel) assignments.

$$\langle (x', y') = (x \times y, x + y) \rangle$$

Use forward substitution and erasure laws.

Solution:

$$\begin{array}{ll} & \langle (x', y') = (x \times y, x + y) \rangle \\ \sqsubseteq & \text{“forward substitution”} \\ & z := x; \langle (x', y') = (z \times y, z + y) \rangle \\ \sqsubseteq & \text{“forward substitution”} \\ & z := x; x : z \times y; \langle (x', y') = (x, z + y) \rangle \\ \sqsubseteq & \text{“erasure”} \\ & z := x; x : z \times y; y := z + y \end{array}$$

Q1[10] Use the alternation law to implement

$$\langle \{x', y'\} = \{x, y\} \wedge x' \leq y' \rangle$$

Solution:

$$\begin{array}{ll} & \langle \{x', y'\} = \{x, y\} \wedge x' \leq y' \rangle \\ \sqsubseteq & \textbf{if } x \leq y \textbf{ then } \langle x \leq y \Rightarrow \{x', y'\} = \{x, y\} \wedge x' \leq y' \rangle \\ & \textbf{else } \langle x > y \Rightarrow \{x', y'\} = \{x, y\} \wedge x' \leq y' \rangle \end{array}$$

The specification for the then part is refined by skip, using the

erausre law. It remains to refine the specification of the else part.

$$
\begin{array}{ll}
 & \langle x > y \Rightarrow \{x', y'\} = \{x, y\} \wedge x' \leq y' \rangle \\
\sqsubseteq & \text{“Strengthening by weakening the antecedant”} \\
 & \langle y \leq x \Rightarrow \{x', y'\} = \{x, y\} \wedge x' \leq y' \rangle \\
\sqsubseteq & \text{“Forward substitution”} \\
 & z := x; \langle y \leq z \Rightarrow \{x', y'\} = \{z, y\} \wedge x' \leq y' \rangle \\
\sqsubseteq & \text{“Forward substitution”} \\
 & z := x; x := y; \langle x \leq z \Rightarrow \{x', y'\} = \{z, x\} \wedge x' \leq y' \rangle \\
\sqsubseteq & \text{“Erasure”} \\
 & z := x; x := y; y := z
\end{array}
$$

Q2.[25] Up edge. Given a constant function $C : \{0, .., N\} \stackrel{\mathsf{tot}}{\rightarrow} \mathbb{B}$, an up edge is an index $i \in \{0, ..N\}$ such that $\neg C(i) \wedge C(i+1)$.To ensure an up edge exists, we will assume $\neg C(0)$ and $C(N)$. Consider the specification

$$f = \langle \mathcal{B} \rangle \text{ where } \mathcal{B} \text{ is } \neg C(i') \wedge C(i'+1)$$

Derive a solution using the binary search technique and the method of invariants. Make sure you use plenty of English prose as well as formal derivation to explain your solution. Here is a suggested outline: Propose an invariant $\mathcal{I}$. Find a command that refines $m = \langle \mathcal{I}' \rangle$. Propose a loop guard $\mathcal{A}$ so that $\mathcal{I} \wedge \neg \mathcal{A} \Rightarrow \widetilde{\mathcal{B}}$ is valid. (Show it is.). Find a loop body that implements $h = \langle \mathcal{A} \wedge \mathcal{I} \Rightarrow \mathcal{I}' \rangle$. (Show that it does.)

Solution:

The proposed invariant $\mathcal{I}$ is $i < k \wedge \neg C(i) \wedge C(k)$. We have (by erasure)

$$\langle i < k \wedge C(i') \wedge C(k') \rangle \sqsubseteq i, k := 0, N$$

The proposed loop guard $\mathcal{A}$ is $k \neq i + 1$. We then have to check that

$$i < k \wedge \neg C(i) \wedge C(k) \wedge k = i + 1 \Rightarrow \neg C(i) \wedge C(i + 1)$$

is valid; by the one-point law, it is.

Finally we need to find a loop bode that implements

$$h = \langle k \neq i + 1 \wedge i < k \wedge \neg C(i) \wedge C(k) \Rightarrow i' < k' \wedge \neg C(i') \wedge C(k') \rangle$$

First note that $k \neq i + 1 \wedge i < k$ simplifies to $k - i > 1$. And, if $k - i > 1$ and $m = \left\lfloor \frac{i+k}{2} \right\rfloor$, then $i < m < k$. Thus we can implement h with

$$
\begin{array}{ll}
 & h \\
\sqsubseteq & \\
 & m := \left\lfloor \dfrac{i + k}{2} \right\rfloor ; \langle i < m < k \wedge \neg C(i) \wedge C(k) \Rightarrow i' < k' \wedge \neg C(i') \wedge C(k') \rangle
\end{array}
$$

Now apply the alternation law to get

$$
\begin{array}{ll}
 & \langle i < m < k \wedge \neg C(i) \wedge C(k) \Rightarrow i' < k' \wedge \neg C(i') \wedge C(k') \rangle \\
\sqsubseteq & \text{“Alternation law”} \\
 & \quad \textbf{if } C(m) \\
 & \quad \textbf{then } \langle m < k \wedge \neg C(i) \wedge C(m) \Rightarrow i' < k' \wedge \neg C(i') \wedge C(k') \rangle \\
 & \quad \textbf{else } \langle i < m \wedge \neg C(m) \wedge C(k) \Rightarrow i' < k' \wedge \neg C(i') \wedge C(k') \rangle \\
\sqsubseteq & \text{“Erasure law for assignment twice.”} \\
 & \quad \textbf{if } C(m) \\
 & \quad \textbf{then } k := m \\
 & \quad \textbf{else } i := m
\end{array}
$$

In summary we have

$$
\begin{array}{ll}
 & \langle \neg C(i') \wedge C(i'+1) \rangle \\
\sqsubseteq & \\
 & \quad i, k := 0, N; \\
 & \quad \text{// Inv.: } i < k \wedge \neg C(i) \wedge C(k) \\
 & \quad \text{// Bound: } k - i \\
 & \quad \textbf{while } k \neq i+1 \\
 & \quad \textbf{do } (\ m := \left\lfloor \tfrac{i+k}{2} \right\rfloor ; \\
 & \qquad \textbf{if } C(m) \\
 & \qquad \textbf{then } k := m \\
 & \qquad \textbf{else } i := m\ )
\end{array}
$$

Finally we should informally check that the loop will terminate. Since $i < m < k$, we can see that both assignments decrease $k - i$, so $k - i$ is a bound.

Q3[25] The log base 2

Use the method of invariants to derive an algorithm for the following specification where x and y are integer variables.

$$\left\langle y > 0 \Rightarrow z' = 2^{x'} \leq y < 2^{x'+1} \right\rangle$$

Solution:

Let the invariant $\mathcal{I}$ be $y > 0 \Rightarrow z = 2^x \leq y$ and the guard $\mathcal{A}$ be $y \geq 2^{x+1}$. Now we need to check that $\mathcal{I} \wedge \neg\mathcal{A} \Rightarrow \left(y > 0 \Rightarrow z = 2^x \leq y < 2^{x+1}\right)$ is valid.

We need initialization code that establishes $\mathcal{I}$ without changing y.

$$
\begin{array}{ll}
 & \mathcal{I}' \wedge y' = y \\
\sqsubseteq & \text{“erasure law”} \\
 & x, z := 0, 1
\end{array}
$$

This is justified by seeing that

$$\big(y > 0 \Rightarrow 1 = 2^0 \leq y\big) \wedge y = y$$

is valid

We need a loop body that preserves the invariant without changing y.

$$\begin{array}{ll} & \langle \mathcal{A} \wedge \mathcal{I} \Rightarrow \mathcal{I}' \wedge y' = y \rangle \\ = & \text{“expanding definitions”} \\ & \Big\langle y \geq 2^{x+1} \wedge (y > 0 \Rightarrow z = 2^x \leq y) \Rightarrow \Big(y > 0 \Rightarrow z' = 2^{x'} \leq y'\Big) \wedge y' = y \Big\rangle \\ \sqsubseteq & \text{“erasure law”} \\ & x, z := x + 1, z + z \end{array}$$

The last step is justified by seening that

$$y \geq 2^{x+1} \wedge (y > 0 \Rightarrow z = 2^x \leq y) \Rightarrow \big(y > 0 \Rightarrow z + z = 2^{x+1} \leq y\big) \wedge y = y$$

is valid.

The work above justifies the following solution

$$\begin{array}{ll} & \Big\langle y > 0 \Rightarrow z' = 2^{x'} \leq y < 2^{x'+1} \Big\rangle \\ \sqsubseteq & \\ & \quad x, z := 0, 1; \\ & \quad \text{// Inv.: } y > 0 \Rightarrow z = 2^x \leq y \\ & \quad \text{// Bound: } y - 2^x \\ & \quad \textbf{while } y \geq 2^{x+1} \\ & \quad \textbf{do } x, z := x + 1, z + z \end{array}$$