Program Logics Hand-in 3

Zijun Yu 202203581

Octobor 2023

Exercise 1

By applying Ht-beta, Ht-bind, and Ht-alloc, it suffices to show

$$\forall v.\{\exists l. v = l \land l \hookrightarrow 0\}\mathsf{let}\, x := v\, \mathsf{in}\, (\lambda\_.x \leftarrow !x+1, \lambda\_.!x)\{\text{The original postcondition}\}$$

We introduce v and apply Ht-let-bind, Ht-exist, ∀I (to introduce l), and EQ, then it suffices to show

$$\{l \hookrightarrow 0\}(\lambda\_.l \leftarrow !l+1, \lambda\_.!l)\{\text{The original postcondition}\}$$

We instantiate C to $\lambda n.l \hookrightarrow n$ and the goal becomes:

$$\{l \hookrightarrow 0\}(\lambda\_.l \leftarrow !l+1, \lambda\_.!l)\left\{ v.\begin{array}{l} l \hookrightarrow 0 * \\ \forall n.\{l \hookrightarrow n\}v.inc()\{u.u = () \land l \hookrightarrow (n+1)\} * \\ \forall n.\{l \hookrightarrow n\}v.read()\{u.u = n \land l \hookrightarrow n\} \end{array} \right\}$$

Then, it suffices to show the following three goals:

$$\begin{array}{l} \{l \hookrightarrow 0\}(\lambda\_.l \leftarrow !l+1, \lambda\_.!l)\{v.l \hookrightarrow 0\} \\ \{True\}(\lambda\_.l \leftarrow !l+1, \lambda\_.!l)\{v.\forall n.\{l \hookrightarrow n\}v.inc()\{u.u = () \land l \hookrightarrow (n+1)\}\} \\ \{True\}(\lambda\_.l \leftarrow !l+1, \lambda\_.!l)\{v.\forall n.\{l \hookrightarrow n\}v.read()\{u.u = n \land l \hookrightarrow n\}\} \end{array}$$

The first one is trivial. The second and the third are similar and we only show the former. For the second one, it suffices to show

$$\forall n.\{l \hookrightarrow n\}\,(\pi_1(\lambda\_.l \leftarrow !l+1, \lambda\_.!l))\,()\{u.u = () \land l \hookrightarrow (n+1)\}$$

By applying ∀I, Ht-bind-det, Ht-proj, and Ht-beta, it suffices to show

$$\{l \hookrightarrow n\}l \leftarrow !l+1\{u.u = () \land l \hookrightarrow (n+1)\}$$

which is straightforward by Ht-bind-det, Ht-load, and Ht-store.

Exercise 2

Item 1

We first do case analysis on listFilter $P\ ys$. It is either [] or some non-empty sequence $y' :: ys'$.

If listFilter $P\ ys \equiv []$, then we have that isList a' $[] \equiv a' = \mathsf{inl}()$. Then by applying Ht-beta, Ht-bind-det, Ht-proj several times, our goal becomes

$$\{a' = \mathsf{inl}()\}\mathsf{if}\, p\ x'\ \mathsf{then}\ \mathsf{inr}(\mathsf{ref}\,(x', \mathsf{inl}()))\ \mathsf{else}\ \mathsf{inl}()\{v.\text{isList}\ v\ (\text{listFilter}\ P\ [x'])\}$$

Then we apply HT-BIND with Q beging instantiated to the postcondition of the assumption $\forall x.\{True\}P\ x\{v.\ \text{isBool}\ v * v = P\ x\}$. We then do case analysis on $P\ x'$. In each case, we apply HT-IF-TRUE and HT-IF-FALSE respectively, and the rest of the proof is straightforward.

In the case where listFilter P ys is not empty, according to the definition of isList, we have that for some hd and l', $a' = \mathsf{inr}(hd) * hd \hookrightarrow (y, l') * isList\ l'\ ys'$. Now again, by applying HT-BETA, HT-BIND-DET, HT-PROJ several times, our goal becomes

$$\begin{aligned}&\{a' = \mathsf{inr}(hd) * hd \hookrightarrow (y, l') * isList\ l'\ ys' * \text{isList}\ \mathsf{inr}(hd)\ (\text{listFilter}\ P\ ys)\}\\ &\quad \mathsf{if}\ p\ x'\ \mathsf{then}\ \mathsf{inr}(\mathsf{ref}\,(x', \mathsf{inr}(hd)))\ \mathsf{else}\ \mathsf{inr}(hd)\\ &\{v.\text{isList}\ v\ (\text{listFilter P (x' :: ys)})\}\end{aligned}$$

We again apply HT-BIND and do case analysis on $P\ x'$. In the case where $P\ x' = \mathsf{true}$, after unfolding listFilter and isList once, we need to show

$$\begin{aligned}&\{a' = \mathsf{inr}(hd) * hd \hookrightarrow (y, l') * isList\ l'\ ys' * \text{isList}\ \mathsf{inr}(hd)\ (\text{listFilter}\ P\ ys)\}\\ &\quad \mathsf{inr}(\mathsf{ref}\,(x', \mathsf{inr}(hd)))\\ &\{v.\exists hd', l'.v = \mathsf{inr}(hd) * hd' \hookrightarrow (x', l') * \text{isList}\ l'\ (\text{listFilter P ys})\}\end{aligned}$$

Then we use HT-BIND to evaluate $\mathsf{ref}\,(x', \mathsf{inr}(hd))$, and we are left to prove two sub-goals:

1.

$$\begin{aligned}&\{a' = \mathsf{inr}(hd) * hd \hookrightarrow (y, l') * isList\ l'\ ys' * \text{isList}\ \mathsf{inr}(hd)\ (\text{listFilter}\ P\ ys)\}\\ &\quad \mathsf{ref}\,(x', \mathsf{inr}(hd))\\ &\{v.\exists hd'.v = hd' \wedge hd' \hookrightarrow (x', \mathsf{inr}(hd)) * a' = \mathsf{inr}(hd) * hd \hookrightarrow (y, l') * isList\ l'\ ys' * \text{isList}\ \mathsf{inr}(hd)\ (\text{listFilter}\ P\ ys)\}\end{aligned}$$

2.

$$\begin{aligned}&\forall v.\{\exists hd'.v = hd' \wedge hd' \hookrightarrow (x', \mathsf{inr}(hd)) * a' = \mathsf{inr}(hd) * hd \hookrightarrow (y, l') * isList\ l'\ ys' * \text{isList}\ \mathsf{inr}(hd)\ (\text{listFilter}\ P\ ys)\}\\ &\qquad \mathsf{inr}(v)\\ &\quad \{v.\exists hd', l'.v = \mathsf{inr}(hd') * hd' \hookrightarrow (x', l') * \text{isList}\ l'\ (\text{listFilter P ys})\}\end{aligned}$$

The first is by HT-ALLOC. For the second one, it suffices to show

$$\begin{aligned}&\{hd' \hookrightarrow (x', \mathsf{inr}(hd)) * a' = \mathsf{inr}(hd) * hd \hookrightarrow (y, l') * isList\ l'\ ys' * \text{isList}\ \mathsf{inr}(hd)\ (\text{listFilter}\ P\ ys)\}\\ &\quad \mathsf{inr}(hd')\\ &\{v.\exists hd', l'.v = \mathsf{inr}(hd') * hd' \hookrightarrow (x', l') * \text{isList}\ l'\ (\text{listFilter P ys})\}\end{aligned}$$

where we instantiate hd' and l' in the postcondition to hd' and $\mathsf{inr}(hd)$ respectively and then it is trivial to show.

Similarly, we can prove this item in the case where $P\ x' = \mathsf{false}$.

Item 2

We do induction on xs. If $xs = []$, it is trivial. In the case where $xs = x :: xs'$, we need to prove $all\ (\lambda x.True)\ (x :: xs')$. It suffices to show that $(\lambda x.True)$, which reduces to $True$, and $all\ (\lambda x.True)\ xs'$, which is exactly the assumption and the induction hypothesis.

Item 3

Plug in the definition, it suffices to show $\mathsf{inl}() = \mathsf{inl}()$, which is true.