Program Logics Hand-in 2

Zijun Yu 202203581

September 2023

Exercise 1

$$\dfrac{S \vdash \{l \hookrightarrow 1\} l \leftarrow 2 \{v. \wedge l \hookrightarrow 2\}}{S \vdash \{l \hookrightarrow 1 \wedge l \hookrightarrow 1\} l \leftarrow 2 \{v.l \hookrightarrow 2 \wedge l \hookrightarrow 1\}} \textsc{Ht-frame-invalid}$$

We got to a point where we have $l \hookrightarrow 1 \wedge l \hookrightarrow 2$, suggesting that the rule is unsound.

Exercise 2

We first prove the following Ht-pre-eq rule:

$$\dfrac{\dfrac{\dfrac{}{\Gamma, x \vdash Q : Prop} \quad \dfrac{\Gamma \mid S \vdash \{P[v/x]\} e[v/x] \{u.Q[v/x]\}}{\Gamma, x \mid S \wedge x = v \vdash \{P[v/x]\} e[v/x] \{u.Q[v/x]\}} \quad \dfrac{}{\Gamma, x \mid S \wedge x = v \vdash x = v}}{\Gamma, x \mid S \wedge x = v \vdash \{P\} e \{u.Q\}} \text{Eq}}{\Gamma, x \mid S \vdash \{x = v \wedge P\} e \{u.Q\}} \textsc{Ht-eq}$$

Then we prove Ht-bind-det:

$$\dfrac{S \vdash \{P\} e \{x.x = u \wedge Q\} \qquad \dfrac{\dfrac{S \vdash \{Q[u/x]\} E[u] \{w.R\}}{x \mid S \vdash \{x = u \wedge Q\} E[x] \{w.R\}} \textsc{Ht-pre-eq}}{S \vdash \forall x. \{x = u \wedge Q\} E[x] \{w.R\}}}{S \vdash \{P\} E[e] \{w.R\}} \textsc{Ht-bind}$$

We cannot use this rule in the case where $e = ref(0)$ as we will not be able to prove the middle antecedent, i.e. $S \vdash \{P\} ref(0) \{x.x = u \wedge Q\}$, because the result of $ref(0)$ is not deterministic and we might have assigned u to a location that is already allocated.

Exercise 3

In this definition of linked list, each node is a pair of the value and either null (injection left) or the pointer (location) to the next node. If we expand the recursive definition of isList, by the seperating conjunction, we are ensured that all the locations are different, hence the list is acyclic.

If we use ordinary conjunction, we can have a cyclic list, for example, let $l = \mathsf{inr}(hd)$ where $hd \hookrightarrow (1, l)$, then l is a cyclic list. And l is a valid input to the predicate isList, since we can choose all the l' to be l when expanding the recursion.

Exercise 4

Part 1

We proceed by the Ht-Rec rule and we consider two cases: we use the fact that the sequence xs is either empty $[]$ or has a head x followed by another sequence xs'.

In the first case we need to show that

$$\forall ys, l, l'.\{\text{isList } l\ [] * \text{isList } l'\ ys\}\mathsf{match}\ l\ \mathsf{with}...\{v.\text{isList } v\ ([] + +ys)\}$$

Because isList l $[] \equiv l = \mathsf{inl}()$, then by Ht-match and Ht-csq, our goal becomes

$$\forall ys, l'.\{l = \mathsf{inl}() * \text{isList } l'\ ys\}l'\{v.\text{istList } v\ ys\}$$

which obviously holds. (To be precise, we need to discuss the two cases of ys again, and in each case, use Ht-pre-eq and Ht-ret.)

In the second case, we need to show that

$$\forall x, xs', ys, l, l'.\{\text{isList } l\ (x :: xs') * \text{isList } l'\ ys\}\mathsf{match}\ l\ \mathsf{with}...\{v.\text{isList } v\ ((x :: xs') + +ys)\}$$

where

$$\text{isList } l\ (x :: xs') \equiv \exists hd, t.l = \mathsf{inr}hd * hd \hookrightarrow (x, t) * \text{isList } t\ xs'$$

Thus we can prove

$$\{l = \mathsf{inr}hd * hd \hookrightarrow (x, t) * \text{isList } t\ xs'\}l\{v.v = \mathsf{inr}hd * l = \mathsf{inr}hd * hd \hookrightarrow (x, t) * \text{isList } t\ xs'\}$$

In this case, the second branch of the match is taken, and by the derived rule from exercise 4.10, we proceed to verify the body of the second branch.

We initially have

$\{l = \mathsf{inr}hd * hd \hookrightarrow (x, t) * \text{isList } t\ xs'\}$.

After processing $\mathsf{let}\ p := !hd\ \mathsf{in}$, we have

$\{p = (x, t) * l = \mathsf{inr}hd * hd \hookrightarrow (x, t) * \text{isList } t\ xs'\}$

After processing $\mathsf{let}\ r := \mathsf{append}\ \mathsf{in}$, using the induction hypothesis introduced by Ht-Rec, we have

$\{\text{isList } r\ (xs' + +ys) * p = (x, t) * l = \mathsf{inr}hd * hd \hookrightarrow (x, t) * \text{isList } t\ xs'\}$

After processing $hd \leftarrow (\pi_1 p, r)$, we have

$\{\text{isList } r\ (xs' + +ys) * p = (x, t) * l = \mathsf{inr}hd * hd \hookrightarrow (x, r) * \text{isList } t\ xs'\}$

and we are left to prove that the value of the $\mathsf{inr}hd$ is the sequence we want, i.e. isList $v\ (xs + +ys)$, which is true since $x :: (xs' + +ys) = (x :: xs') + +ys = xs + +ys$.

Part 2

No. Because if we use this specification on $\mathsf{append}\ l\ l$, we will get a cyclic list as the result, and we will not be able to prove isList $v\ (xs + +xs)$.