Alexander J. Summers

Program Verification

Exercise Solutions 9: Heap Reasoning and Permissions

Assignment 1 (Pure Assertions)

1. By (structural) induction on A.

   (Case A is e for some expression e:) Then we have:

$$\begin{aligned} H, P, \sigma \vDash A &\Leftrightarrow \lceil e \rfloor = \textit{true} \\ &\Leftrightarrow H, \varnothing, \sigma \vDash A \end{aligned}$$

   (Case A is $A_1 * A_2$ for some A_1 and A_2:) Then we have:

$$\begin{aligned} H, P, \sigma \vDash A &\Leftrightarrow \exists P_1, P_2.\ P = P_1 \uplus P_2 \textit{ and } H, P_1, \sigma \vDash A_1 \textit{ and } H, P_2, \sigma \vDash A_2 \\ &\Rightarrow H, \varnothing, \sigma \vDash A_1 \textit{ and } H, \varnothing, \sigma \vDash A_2 \textit{ (by induction hypothesis, twice)} \\ &\Rightarrow H, \varnothing, \sigma \vDash A_1 * A_2 \end{aligned}$$

   All other cases follow analogously, by a straightforward induction argument.

2. To prove equivalence, we need to show, for all such A' and A that: $\forall H, P, \sigma.(H, P, \sigma \vDash A * A' \Leftrightarrow H, P, \sigma \vDash A \wedge A')$. We show the $\Rightarrow$ and $\Leftarrow$ directions of this property, for arbitrary such A' and (pure) A, as follows:

   ($\Rightarrow$:) To show this direction of the result, we need an additional lemma, effectively stating that increasing the permissions held in a state will never make assertions false (this result was discussed in the lecture). If we use $P_1 \sqsubseteq P_2$ to mean that P_2 has at least as much permission as P_1 for all locations, then lemma can be stated as follows:

$$\forall A, H, P_1, P_2, \sigma.(\textit{if } H, P_1, \sigma \vDash A \textit{ and } P_1 \sqsubseteq P_2 \textit{ then } H, P_2, \sigma \vDash A)$$

   This lemma can be proved by straightforward induction on A. Using the lemma, we can now show the intended result:
   Let H,P,σ be arbitrary, and assume $H, P, \sigma \vDash A * A'$. Then, by definition, there are some P_1 and P_2 such that: $P = P_1 \uplus P_2$ *and* $H, P_1, \sigma \vDash A$ *and* $H, P_2, \sigma \vDash A'$. Note that, $P_1 \sqsubseteq P$ and $P_2 \sqsubseteq P$. Therefore, by the lemma above, we have $H, P, \sigma \vDash A$ *and* $H, P, \sigma \vDash A'$, and thus, $H, P, \sigma \vDash A \wedge A'$, as required.

   ($\Leftarrow$:) Let H,P,σ be arbitrary, and assume $H, P, \sigma \vDash A \wedge A'$. By definition, $H, P, \sigma \vDash A$ and $H, P, \sigma \vDash A'$. By part (1), we have $H, \varnothing, \sigma \vDash A$. Therefore, since $\varnothing \uplus P = P$, we have $H, P, \sigma \vDash A * A'$, as required.

Assignment 2 (Permissions Required by an Assertion)

1. Imagine we have an assertion $b \Rightarrow$ *acc*$(x.f, 1)$ where b is a boolean variable. Now if σ maps b to true, then it is clear that the permission mask must map (x, f) to 1. However, if σ maps b to false, the permission mask must map (x, f) to 0 because the function *Perms* is required to return a **minimal** mask. For the same reason, the function *Perms* should also depend on H.

2. The separating conjunction $A * B$ expresses that the permissions required by A are disjoint from the permissions required by B. This means that *Perms*$(A * B)_{(H,\sigma)}$ must return enough permission so that it can be split to satisfy the requirements of A and B separately. However, $A \wedge B$ requires only to have enough permission to satisfy both of them together. As a result, while *acc*$(x.f, 1/2) *$ *acc*$(x.f, 1/2)$ requires full permission to $x.f$, *acc*$(x.f, 1/2) \wedge$ *acc*$(x.f, 1/2)$ can be satisfied with a permission mask that provides only $1/2$ to $x.f$.

3. *Perms*$(A)_{(H,\sigma)}$ defined by cases of A would be:

   Perms$(e)_{(H,\sigma)} = \varnothing$ Here $\varnothing$ is a permission mask that maps all (object, field-name) pairs to 0.
   Perms$(A \wedge B)_{(H,\sigma)} = \max($*Perms*$(A)_{(H,\sigma)},$ *Perms*$(A)_{(H,\sigma)})$
   Here $\max(M_1, M_2)$ returns a pointwise maximum of both maps.
   Perms$(A * B)_{(H,\sigma)} =$ *Perms*$(A)_{(H,\sigma)} \uplus$ *Perms*$(A)_{(H,\sigma)}$
   Here $\uplus$ denotes a pointwise map addition.
   Perms$(e \Rightarrow A)_{(H,\sigma)} =$ *Perms*$(A)_{(H,\sigma)}$ if $\ulcorner e \lrcorner_{(H,\sigma)} =$ *true*
   Here $\ulcorner e \lrcorner_{(H,\sigma)}$ represents the expression evaluation.
   Perms$(e \Rightarrow A)_{(H,\sigma)} = \varnothing$ if $\ulcorner e \lrcorner_{(H,\sigma)} =$ *false*
   Perms$($*acc*$(e.f, p))_{(H,\sigma)} =$ *Map*$((\ulcorner e \lrcorner_{(H,\sigma)}, f) \Rightarrow p)$