Exercise 1. Given that $(p : \mathsf{loc})$ and $(x : \mathsf{Val})$ and $(X : A)$ for some A, in

$$\begin{array}{lcl} p \rightsquigarrow \mathsf{Mlistof}\, R\, L & \equiv & \mathsf{match}\, L\, \mathsf{with} \\ & & \mid \mathsf{nil} \Rightarrow [p = \mathsf{null}] \\ & & \mid X :: L' \Rightarrow \exists x p'.\quad p \rightsquigarrow \{\!| \mathrm{hd}{=}x;\ \mathrm{tl}{=}p' |\!\} \\ & & \qquad\qquad\qquad \star\ p' \rightsquigarrow \mathsf{Mlistof}\, R\, L' \\ & & \qquad\qquad\qquad \star\ x \rightsquigarrow R\, X \end{array}$$

Give the type of R:
Give the type of Mlistof:

Exercise 2. Recall that:

$$\begin{array}{lcl} p \rightsquigarrow \mathsf{MList}\, L & \equiv & \mathsf{match}\, L\, \mathsf{with} \\ & & \mid \mathsf{nil} \Rightarrow [p = \mathsf{null}] \\ & & \mid x :: L' \Rightarrow \exists p'.\quad p \rightsquigarrow \{\!| \mathrm{hd}{=}x;\ \mathrm{tl}{=}p' |\!\} \ \star\ p' \rightsquigarrow \mathsf{MList}\, L' \end{array}$$

Define the identity representation predicate Id such that $p \rightsquigarrow \mathsf{Mlistof}\, \mathsf{Id}\, L = p \rightsquigarrow \mathsf{MList}\, L$.

$$\mathsf{Id} \equiv$$

Exercise 3. specify functions over queues using a higher-order representation predicate written $p \rightsquigarrow \mathsf{Queueof}\, R\, L$. Shorthand: just write "$\mathsf{Q}\, R$" instead of "$\mathsf{Queueof}\, R$".

$\{\qquad\qquad\qquad\qquad\}\ (\texttt{create()})\ \{\qquad\qquad\qquad\qquad\qquad\qquad\}$

$\{\qquad\qquad\qquad\qquad\}\ (\texttt{push}\, x\, p)\ \{\qquad\qquad\qquad\qquad\qquad\}$

$\{\qquad\qquad\qquad\qquad\}\ (\texttt{pop}\, p)\ \{\qquad\qquad\qquad\qquad\qquad\qquad\}$

$\{\qquad\qquad\qquad\qquad\}\ (\texttt{concat}\, p\, p')\ \{\qquad\qquad\qquad\qquad\qquad\qquad\}$

Exercise 4. specify a function $\texttt{copy}\, f\, p$ that duplicables a mutable queue specified using Queueof, where f is a function to duplicate items.

$$\begin{array}{l} \big(\forall x X.\ \{\qquad\qquad\}\ (f\, x)\ \{\qquad\qquad\qquad\qquad\}\big) \\ \Rightarrow \{\qquad\qquad\qquad\}\ (\texttt{copy}\, f\, p)\ \{\qquad\qquad\qquad\qquad\qquad\qquad\} \end{array}$$

Exercise 5. rewrite the specification of Mlistof using MCellof.

$$p \rightsquigarrow \mathsf{MCellof}\, R_1\, V_1\, R_2\, V_2 \;\equiv\; \exists v_1 v_2.\quad p \rightsquigarrow \{|\mathrm{hd}{=}v_1;\ \mathrm{tl}{=}v_2|\} \;\star\; v_1 \rightsquigarrow R_1\, V_1 \;\star\; v_2 \rightsquigarrow R_2\, V_2$$

$$p \rightsquigarrow \mathsf{Mlistof}\, R\, L \;\equiv\; \mathsf{match}\ L\ \mathsf{with}\ \mid \mathsf{nil} \Rightarrow [p = \mathsf{null}]\ \mid X :: L' \Rightarrow$$

Exercise 6. rewrite the specification of Narytreeof using Nodeof.

$$p \rightsquigarrow \mathsf{Narytreeof}\, R\, T \equiv \mathsf{match}\ T\ \mathsf{with}\ \mid \mathsf{Leaf} \Rightarrow [p = \mathsf{null}]\ \mid \mathsf{Node}\, X\ L \Rightarrow$$

Exercise 7. complete the specification of Bagof using Nodeof. Hint: chunks are described by the predicate $p' \rightsquigarrow \mathsf{Chunkof}\, R\, E'$.

$$p \rightsquigarrow \mathsf{Bagof}\, R\, T \equiv \mathsf{match}\ T\ \mathsf{with}\ \mid \mathsf{Empty} \Rightarrow [p = \mathsf{null}]\ \mid \mathsf{Layer}\, E'\, T' \Rightarrow$$

Exercise 8. specify the function `miter`, using an invariant of the form $J\,K\,K'$, describing the state before and the state after the iteration.

$$\forall f p R L J.\quad \Big(\forall x X \qquad\qquad .\ \{x \rightsquigarrow R\, X \qquad\qquad\}\ (f\, x)\ \{\lambda\_. \qquad\qquad\}\Big)$$
$$\Rightarrow\ \{p \rightsquigarrow \mathsf{Mlistof}\, R\, L \,\star \qquad\qquad\}\ (\mathtt{miter}\, f\, p)\ \{\lambda\_. \qquad\qquad\}$$

Exercise 9. using the representation predicates Ref (i.e. $x \rightsquigarrow \mathsf{Ref}\, X \equiv x \mapsto X$) and Mlistof, specify the function `(fun x -> incr x)` and `incr_all`. What is $J\ K\ K'$?

```
let incr_all p = miter (fun x -> incr x) p

let example_p = { hd = ref 5; tl = { hd = ref 3; tl = null } }
```

$$\{ \qquad\qquad\qquad \}\ (\texttt{incr x})\ \{\lambda\_. \qquad\qquad\qquad \}$$

$$\{ \qquad\qquad\qquad \}\ (\texttt{incr\_all}\, p)\ \{\lambda\_. \qquad\qquad\qquad \}$$

$J\ K\ K' =$

Exercise 10. Describe the state at the front of each lines (except 5 and 6). Explicit the instantiation of the existential in the invariant.

```
1   let r = ref 0
2   let s = ref n
3   let p = create_lock()
4
5   let concurrent_step () =
6     let () = acquire_lock
          p in
7     incr r;
8     decr s;
9     release_lock p
```

Exercise 11. state a conversion rule relating $p \rightsquigarrow \mathsf{Cellsof}\, R\, M$ with a predicate of the form $p \rightsquigarrow \mathsf{Cellsof}\, \mathsf{Id}\, M'$. Hint: $(R : A \rightarrow a \rightarrow \mathsf{Hprop})$ and $(M : \mathsf{map\, int}\, A)$ and $(M' : \mathsf{map\, int}\, a)$.

$$p \rightsquigarrow \mathsf{Cellsof}\, R\, M \quad =$$

Exercise 12. Let program be:

```
        let r = ref 0
        let r1 = ref 0
        let r2 = ref 0
        let p = create_lock()
acquire_lock p;     ||   acquire_lock p;
r := !r + 1;        ||   r := !r + 1;
r1 := !r1 + 1;      ||   r2 := !r2 + 1;
release_lock p;     ||   release_lock p;
        acquire_lock p;
        assert (!r == 2);
```

Give a lock invariant that allows proving $\{\mathsf{True}\}$ program $\{\mathsf{True}\}$, then prove the triple.