Exercise Set 8: Abstract Datatypes

The PVS file `exercises/abstract_datatypes.pvs` support these exercises. Prove the following lemmas using induction on lists. For some of these, you may need to rewrite with earlier results in the theory. One way to automate this kind of rewriting is to begin each proof with the command `(auto-rewrite-theory "abstract_datatypes")`. Any rewrites that apply will be automatically invoked whenever an assert command is given. Alternatively, you may use either the `:theories` or `:rewrite` options with the PVS proof command `(induct-and-simplify ...)`.

1. The append of a list `l` and the empty list `null` is equal to `l`:

   ```
   append_null: LEMMA append(l, null) = l
   ```

2. Append is transitive:

   ```
   append_assoc: LEMMA
     append(l1, append(l2, l3)) = append(append(l1, l2), l3)
   ```

3. The reverse of append of lists `l1` and `l2` is equal to the append of the reverse of `l2` and the reverse of `l1`:

   ```
   reverse_append: LEMMA
     reverse(append(l1, l2)) = append(reverse(l2), reverse(l1))
   ```

   Hint: Try rewriting with the above two lemmas.

4. The reverse of the reverse of a list `l` is equal to `l`:

   ```
   reverse_reverse: LEMMA
      reverse(reverse(l)) = l
   ```