Idea

Lists are sorted collections of elements, which support insertions and deletions of elements. Elements are retrieved via indexes, which usually start at 0.

Definition

We define the ADT as the following 5-Tuple:

$$\mathcal{D} = (N, P, Fs, Ts, Ax),$$

where the components are defined as follows:

1. $N :=$ List
2. $P :=$ {Element}
3. $Fs :=$ {list, length, isEmpty, get, set, insert, remove}
4. Ts is the set containing the following type specifications:
   (a) list : List
   (b) length : List $\to \mathbb{N}_0$
   (c) isEmpty : List $\to \mathbb{B}$
   (d) get : List $\times \mathbb{N}_0 \to$ Element $\cup\{\Omega\}$
   (e) set : List $\times$ Element $\times \mathbb{N}_0 \to$ List $\cup\{\Omega\}$
   (f) insert : List $\times$ Element $\times \mathbb{N}_0 \to$ List
   (g) remove : List $\times \mathbb{N}_0 \to$ List $\cup\{\Omega\}$
5. Ax is the set containing the following axioms.
   $\forall L \in$ List : $x \in$ Element : $i \in \mathbb{N}_0$:
   (a) list().length() = 0
   (b) list().isEmpty() $\iff$ list().length() = 0
   (c) L.length() $> i \to$ L.set(i, x).get$(i) = x$
   (d) L.length() $\geq i \to$ L.insert(i, x).length() = L.length() + 1
   (e) L.length() $\geq i \to$ L.insert(i, x).get$(i) = x$
   (f) L.length() $\geq i \to$ L.insert(i, x).remove(i) = L
   (g) L.length() $\leq i \to$ L.get$(i) = \Omega$
   (h) L.length() $\leq i \to$ L.set$(i, x) = \Omega$
   (i) L.length() $< i \to$ L.insert$(i, x) = \Omega$
   (j) L.length() $< i \to$ L.remove$(i) = \Omega$

Implementation

TBD