Idea

The idea is to map specific keys to values. A map must thus be able to retrieve the value associated with some key, insert/overwrite the value associated with some key and delete keys.

There are several abstract data types that implement this ADT with further specifications.

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. $N :=$ Map
2. $P :=$ {Key, Value}
3. $Fs :=$ {map, find, insert, delete}
4. Ts is the set containing the following type specifications:
   (a) map : Map
   (b) find : Map $\times$ Key $\rightarrow$ Value $\cup\{\Omega\}$
   (c) insert : Map $\times$ Key $\times$ Value $\rightarrow$ Map $\cup\{\Omega\}$
   (d) find : Map $\times$ Key $\rightarrow$ Map
5. Ax is the set containing the following axioms.
   $\forall L \in$ List : $x \in$ Element : $i \in \mathbb{N}_0$:
   (a) map().find(k) = Ω
   (b) m.insert(k, v).find(k) = v
   (c) $k_1 \neq k_2 \rightarrow m$.insert($k_1, v$).find($k_2$) = m.find(k_2)
   (d) m.delete(k).find(k) = Ω
   (e) $k_1 \neq k_2 \rightarrow m$.delete($k_1$).find($k_2$) = m.find(k_2)

Implementation

TBD