Maps

```
type Map[K,V]

// Constructors
def Empty[K,V]  :  Map[K,V]
def Update[K,V](Map[K,V],  K,  V)  :  Map[K,V]

// Functions
def get(Map[K,V],  K)  :  V

// Predicates
def has(Map[K,V],  K)
```

$$\forall k : \mathrm{K}. \quad \neg \mathrm{has}\left(\mathrm{Empty}\left(\right), k\right) \tag{1}$$

$$\begin{aligned} &\forall m : \mathrm{Map[K,V]},\ k_1, k_2 : \mathrm{K}, v : \mathrm{V}. \\ &\qquad \mathrm{has}\left(\mathrm{Update}\left(m, k_1, v\right), k_2\right) \leftrightarrow \left(k_1 = k_2 \lor \left(k_1 \neq k_2 \land \mathrm{has}\left(m, k_2\right)\right)\right) \end{aligned} \tag{2}$$

$$\begin{aligned} &\forall m : \mathrm{Map[K,V]},\ k_1, k_2 : \mathrm{K}, v : \mathrm{V}. \\ &\qquad \left(k_1 \neq k_2 \rightarrow \mathrm{get}\left(\mathrm{Update}\left(m, k_1, v\right), k_2\right) = \mathrm{get}\left(m, k_2\right)\right) \\ &\qquad \land \left(k_1 = k_2 \rightarrow \mathrm{get}\left(\mathrm{Update}\left(m, k_1, v\right), k_2\right) = v\right) \end{aligned} \tag{3}$$