The Dictionary ADT: values and operations

Looks like: \{a:6, b:7, c:8\}

Signature:

```python
interface DICT[K, V]:
    def mem?(self, key: K) -> bool?
    def get(self, key: K) -> V
    def put(self, key: K, value: V): NoneC
    def del(self, key: K): NoneC
    def empty?(self): bool?
```
The Dictionary ADT: laws

\[
\{\} \emptyset() \Rightarrow \text{True} \}
\]

\[
\{k_0:v_0,\ldots,k_n,v_n\} \emptyset() \Rightarrow \text{False} \}
\]

\[
\{k_0:v_0,\ldots,k_i,v_i,\ldots\} \text{mem?}(k_i) \Rightarrow \text{True} \}
\]

\[
\forall i, k_i \neq k \{\ldots,k_i:v_i,\ldots\} \text{mem?}(k) \Rightarrow \text{False} \}
\]

\[
\{\ldots,k_i:v_i,\ldots\} \text{get}(k_i) \Rightarrow v_i \}
\]

\[
d = \{\ldots,k_i:v_i,\ldots\} \text{d.put}(k_i,v) \Rightarrow \text{None} \quad d = \{\ldots,k_i:v,\ldots\}
\]

\[
d = \{\ldots,k_i:v_i,\ldots\} \land \forall i, k_i \neq k \text{d.put}(k,v) \Rightarrow \text{None} \quad d = \{\ldots,k_i:v_i,\ldots,k:v\}
\]

\[
d = \{\ldots,k_i:v_i,\ldots\} \text{d.del}(k_i) \Rightarrow \text{None} \quad d = \{\ldots,k_{i-1}:v_{i-1},k_{i+1}:v_{i+1},\ldots\}
\]

\[
d = \{\ldots,k_i:v_i,\ldots\} \land \forall i, k_i \neq k \text{d.del}(k) \Rightarrow \text{None} \quad d = \{\ldots,k_i:v_i,\ldots\}
\]
Law breakdown: \( \text{mem?} \)

If the key we are looking for is present, we get true:

\[
\{ \} \{ k_0:v_0, \ldots, k_i:v_i, \ldots \} . \text{mem?}(k_i) \Rightarrow \text{True} \ \{ \}
\]

If the key we are looking for is not equal to any of the keys in the dictionary, we get false:

\[
\{ \forall i, k_i \neq k \} \{ \ldots, k_i:v_i, \ldots \} . \text{mem?}(k) \Rightarrow \text{False} \ \{ \}
\]
Law breakdown: *get*

If we try to lookup a key present in the dictionary, we get its associated value:

\[
\{ \ldots, k_i:v_i, \ldots \} \cdot \text{get}(k_i) \Rightarrow v_i \quad \{
\]

If we try to lookup a key that isn’t among the dictionary’s keys—that’s the precondition \( k \neq k_i \)—then there is not result. It may be an error, but different APIs in different languages will indicate this in different ways.
Law breakdown: *put*

If we put a key that’s already present, its associated value gets replaced:

\[
\begin{align*}
\{d = \{\ldots, k_i:v_i, \ldots\}\} \\
&\text{\texttt{d.put}(k_i, v) \Rightarrow None} \\
\{d = \{\ldots, k_i:v, \ldots\}\}
\end{align*}
\]

If we put a key that’s absent, the new key and value association is added:

\[
\begin{align*}
\{d = \{\ldots, k_i:v_i, \ldots\}\} \wedge \forall i, k_i \neq k \\
&\text{\texttt{d.put}(k, v) \Rightarrow None} \\
\{d = \{\ldots, k_j:v_i, \ldots, k:v\}\}
\end{align*}
\]
Law breakdown: \textit{del}

If we delete a key that’s present, it gets removed:

\[
\{d = \{\ldots, k_i:v_i, \ldots \}\}\]

\[d.\text{del}(k_i) \Rightarrow \text{None}\]

\[
\{d = \{\ldots, k_{i-1}:v_{i-1}, k_{i+1}:v_{i+1}, \ldots \}\}\]

If we delete a key that’s absent, nothing happens:

\[
\{d = \{\ldots, k_i:v_i, \ldots \}\} \land \forall i, k_i \neq k\]

\[d.\text{del}(k) \Rightarrow \text{None}\]

\[
\{d = \{\ldots, k_i:v_i, \ldots \}\}\]
Next: a data structure for dictionaries