The Dictionary ADT

CS 214, Fall 2019

The Dictionary ADT: values and operations

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

Signature:

```
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

$$\{\}\ \boxed{\{\}}.\mathit{empty?}() \Rightarrow \texttt{True}\ \{\}$$

$$\{\}\ \boxed{\{k_0{:}v_0, \ldots, k_n, v_n\}}.\mathit{empty?}() \Rightarrow \texttt{False}\ \{\}$$

$$\{\}\ \boxed{\{k_0{:}v_0, \ldots, k_i{:}v_i, \ldots\}}.\mathit{mem?}(k_i) \Rightarrow \texttt{True}\ \{\}$$

$$\{\forall i, k_i \neq k\}\ \boxed{\{\ldots, k_i{:}v_i, \ldots\}}.\mathit{mem?}(k) \Rightarrow \texttt{False}\ \{\}$$

$$\{\}\ \boxed{\{\ldots, k_i{:}v_i, \ldots\}}.\mathit{get}(k_i) \Rightarrow v_i\ \{\}$$

$$\left\{d = \boxed{\{\ldots, k_i{:}v_i, \ldots\}}\right\}\ d.\mathit{put}(k_i, v) \Rightarrow \texttt{None}\ \left\{d = \boxed{\{\ldots, k_i{:}v, \ldots\}}\right\}$$

$$\left\{d = \boxed{\{\ldots, k_i{:}v_i, \ldots\}} \wedge \forall i, k_i \neq k\right\}\ d.\mathit{put}(k, v) \Rightarrow \texttt{None}\ \left\{d = \boxed{\{\ldots, k_i{:}v_i, \ldots, k{:}v\}}\right\}$$

$$\left\{d = \boxed{\{\ldots, k_i{:}v_i, \ldots\}}\right\}\ d.\mathit{del}(k_i) \Rightarrow \texttt{None}\ \left\{d = \boxed{\{\ldots, k_{i-1}{:}v_{i-1}, k_{i+1}{:}v_{i+1}, \ldots\}}\right\}$$

$$\left\{d = \boxed{\{\ldots, k_i{:}v_i, \ldots\}} \wedge \forall i, k_i \neq k\right\}\ d.\mathit{del}(k) \Rightarrow \texttt{None}\ \left\{d = \boxed{\{\ldots, k_i{:}v_i, \ldots\}}\right\}$$

Law breakdown: *mem?*

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

$$\{\} \; \boxed{\{k_0{:}v_0, \ldots, k_i{:}v_i, \ldots\}}.mem?(k_i) \Rightarrow \texttt{True} \; \{\}$$

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

$$\{\forall i, k_i \neq k\} \; \boxed{\{\ldots, k_i{:}v_i, \ldots\}}.mem?(k) \Rightarrow \texttt{False} \; \{\}$$

Law breakdown: *get*

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

$$\{\}\ \boxed{\{\ldots, k_i{:}v_i, \ldots\}}.get(k_i) \Rightarrow v_i\ \{\}$$

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:

$$\left\{d = \boxed{\{\ldots, k_i{:}v_i, \ldots\}}\right\}$$

$$d.put(k_i, v) \Rightarrow \text{None}$$

$$\left\{d = \boxed{\{\ldots, k_i{:}v, \ldots\}}\right\}$$

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

$$\left\{d = \boxed{\{\ldots, k_i{:}v_i, \ldots\}} \land \forall i, k_i \neq k\right\}$$

$$d.put(k, v) \Rightarrow \text{None}$$

$$\left\{d = \boxed{\{\ldots, k_i{:}v_i, \ldots, k{:}v\}}\right\}$$

Law breakdown: *del*

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

$$\left\{d = \boxed{\{\ldots, k_i{:}v_i, \ldots\}}\right\}$$

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

$$\left\{d = \boxed{\{\ldots, k_{i-1}{:}v_{i-1}, k_{i+1}{:}v_{i+1}, \ldots\}}\right\}$$

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

$$\left\{d = \boxed{\{\ldots, k_i{:}v_i, \ldots\}} \land \forall i, k_i \neq k\right\}$$

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

$$\left\{d = \boxed{\{\ldots, k_i{:}v_i, \ldots\}}\right\}$$

Next: a data structure for dictionaries