

0.0.1 Update

The operation *update* will return a Collection *coll!* which is the same as the input Collection *coll?* except for at index *idx?*. The existing member *coll?*_{*idx?*} is replcaed by the provided Value *v?* at *idx?* in *coll!* such that

$$idx? \mapsto v? \in coll! \wedge idx? \mapsto coll?_{idx?} \notin coll!$$

which is equivalent to *remove* \gg *append*

$$update(coll?, v?, idx?) \equiv append(remove(coll?, idx?), v?, idx?)$$

The functionality of *update* is further explained in the following schema.

<div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="margin-bottom: 10px;"> $Update[Collection, V, \mathbb{N}]$ </div> <div style="border-bottom: 1px solid black; width: 90%;"></div> </div> <div style="margin-bottom: 10px;"> $idx? : \mathbb{N}$ $coll?, coll! : Collection$ $v? : V$ $update_ : Collection \times V \times \mathbb{N} \rightsquigarrow Collection$ </div> <hr style="border: 0.5px solid black; margin: 5px 0;"/> <div> $1 = \# idx?$ $coll! = update(coll?, v?, idx?) \bullet$ $let coll' == \{i : \mathbb{N} \mid i \in 0..idx?\} \upharpoonright coll?$ $coll'' == head(coll') \cap v?$ $coll''' == \{j : \mathbb{N} \mid j \in idx? + 1 .. \# coll?\} \upharpoonright coll?$ $= coll'' \cap coll''' \Rightarrow$ $(append(remove(coll', idx?), v?, idx?) \cap coll'') \wedge$ $(v? \mapsto idx? \in coll!) \wedge$ $(\# coll! = \# coll?) \wedge$ </div>
--

The value which previously existed at $idx? \in coll?$ is replaced with *v?* to result in *coll!*

- *coll'* is the items in *coll?* up to and including *idx?*
- *coll''* is the items in *coll?* except the item at *idx?* has been replaced with *v?*
- *coll'''* is the items in *coll?* from *idx?* + 1 to $\# coll? \Rightarrow coll?_{idx?} \notin coll''$

The following example illustrates these properties.

$$\begin{aligned}
X &= \langle x_0, x_1, x_2 \rangle \\
x_0 &= 0 \\
x_1 &= foo \\
x_2 &= \langle a, b, c \rangle \\
v? &= bar
\end{aligned}$$

$update(X, v?, 0) = \langle bar, foo, \langle a, b, c \rangle \rangle$
 $update(X, v?, 1) = \langle 0, bar, \langle a, b, c \rangle \rangle$
 $update(X, v?, 2) = \langle 0, foo, bar \rangle$
 $update(X, v?, 3) = \langle 0, foo, \langle a, b, c \rangle, bar \rangle$
 $update(X, v?, 4) = append(X, v?, 3) = update(X, v?, 3) \iff 3 \notin \text{dom } X$