0.0.1 Map

The *map* operation takes in a function $fn?$, Collection $coll?$ and additional Arguments $args?$ (as necessary) and returns a modified Collection $coll!$ with members $fn!_n$. The ordering of $coll?$ is maintained within $coll!$

$$
\begin{array}{|l}
Map[(\_ \pfun \_), Collection, V] \\
\hline
fn? : (\_ \pfun \_) \\
args? : V \\
coll? , coll! : Collection \\
map\_ : (\_ \pfun \_) \times Collection \times V \pfun Collection \\
\hline
coll! = map(fn? , coll? , args?) \bullet \\
\qquad \langle \forall n : i \,..\, j \in coll? \mid i \leq n \leq j \land j = \# coll? \bullet \\
\qquad\quad \exists_1 fn!_n : V \mid fn!_n = \\
\qquad\qquad (fn? (coll?_n , args?) \iff args? \neq \emptyset) \lor \\
\qquad\qquad (fn? (coll?_n) \iff args? = \emptyset) \rangle \Rightarrow fn!_i \frown fn!_n \frown fn!_j \\
\hline
\end{array}
$$

Above, $fn!_n$ is introduced to handle the case where $fn?$ only requires a single argument. Additional arguments may be necessary but if they are not ($args? = \emptyset$) then only $coll?_n$ is passed to $fn?$.

$$
\begin{array}{ll}
X = \langle 1, 2, 3 \rangle & \\
\quad map\,(succ, X) = \langle 2, 3, 4 \rangle & \text{[increment each member of } X\text{]} \\
\quad map\,(+, X, 2) = \langle 3, 4, 5 \rangle & \text{[add 2 to each member of } X\text{]}
\end{array}
$$