0.1 Map

The *map* operation accepts the following arguments

- Operation or Primitive
- A Collection
- Additional Arguments passed to the Operator or Primitive

and returns a collection of Operation $Operation(x_n, args) \vee Primitive(x_n, args)$ respectively

Given an input Collection X and a Operation o or Primitive p where

$$X =< x_i..x_n..x_j >$$

and that collection consists of one or more members x_n within the range $i..j$

$$i \leq n \leq j \Rightarrow i \prec n \prec j \iff i \neq n \neq j$$

then

$$map(o,\ X,\ args) = Y\ \wedge\ map(p,\ X,\ args) = Y'$$

such that

$$Y =< o(x_i, args)..o(x_n, args)..o(x_j, args) >$$

and

$$Y' =< p(x_i, args)..p(x_n, args)..p(x_j, args) >$$

which establishes both Y and Y' are a Collection where each member y_n or y'_n is the result of passing x_n and $args$ to o or p respectively.

In otherwords

$$o(x_i, args) \mapsto y_i \wedge o(x_n, args) \mapsto y_n \wedge o(x_j, args) \mapsto y_j$$

$$p(x_i, args) \mapsto y'_i \wedge p(x_n, args) \mapsto y'_n \wedge p(x_j, args) \mapsto y'_j$$

which implies both collections Y and Y' have the same ordering as collection X

$$i_{o(x,args)} = i_y\ \wedge\ n_{o(x,args)} = n_y\ \wedge\ j_{o(x,args)} = j_y$$

$$i_{p(x,args)} = i_{y'}\ \wedge\ n_{p(x,args)} = n_{y'}\ \wedge\ j_{p(x,args)} = j_{y'}$$

When X contains non-distinct values, o and p are unaffected.

$$o(x_n, args) = y_n$$

$$o(x_{n'}, args) = y_{n'}$$

$$o(x_{n'+1}, args) = y_{n'}$$

$$\iff$$

$$x_{n'} \equiv x_{n'+1} \land x_{n'} \not\equiv x_n$$

$$\Rightarrow$$

$$x_n \not\equiv x_{n'+1}$$

$$\Rightarrow$$

$$o(x_{n'}, args) = o(x_{n'+1}, args) \neq o(x_n, args)$$

Because p is just a composition of o's, the same property holds for primitives

$$p(x_n, args) = y'_n$$

$$p(x_{n'}, args) = y'_{n'}$$

$$p(x_{n'+1}, args) = y'_{n'}$$

$$\iff$$

$$x_{n'} \equiv x_{n'+1} \land x_{n'} \not\equiv x_n$$

$$\Rightarrow$$

$$x_n \not\equiv x_{n'+1}$$

$$\Rightarrow$$

$$p(x_{n'}, args) = p(x_{n'+1}, args) \neq p(x_n, args)$$