

Basic Generic Definitions

Example 1 : Generic Identity Function

The identityGB function works for any type:

$$\begin{array}{c} \boxed{[X]} \\ \hline \boxed{\begin{array}{l} \text{identityGB} : X \rightarrow X \\ \forall x : X \bullet \text{identityGB}(x) = x \end{array}} \end{array}$$

This defines identityGB polymorphically. It can be used with any type: identityGB applied to a natural number returns that number, identityGB applied to a set returns that set, etc.

Example 2 : Generic Constant

A generic empty set:

$$\begin{array}{c} \boxed{[X]} \\ \hline \boxed{\begin{array}{l} \text{empty} : \mathbb{P} X \\ \text{empty} = \{\} \end{array}} \end{array}$$

The empty set can be instantiated for any type X.

Example 3 : Generic Pair Functions

Extract the first element of a pair:

$$\begin{array}{c} \boxed{[X, Y]} \\ \hline \boxed{\begin{array}{l} \text{fst} : X \times Y \rightarrow X \\ \forall x : X \bullet \forall y : Y \bullet \text{fst}(x, y) = x \end{array}} \end{array}$$

Extract the second element:

$$\begin{array}{c} \boxed{[X, Y]} \\ \hline \boxed{\begin{array}{l} \text{snd} : X \times Y \rightarrow Y \\ \forall x : X \bullet \forall y : Y \bullet \text{snd}(x, y) = y \end{array}} \end{array}$$

These work for pairs of any types.

Example 4 : Generic Singleton Set

Create a set containing a single element:

$$\begin{array}{c} \boxed{[X]} \\ \hline \boxed{\begin{array}{l} \text{singleton} : X \rightarrow \mathbb{P} X \\ \forall x : X \bullet \text{singleton}(x) = \{x\} \end{array}} \end{array}$$

Given any value, singleton produces a set containing just that value.

Example 5 : Generic Sequence Operations

The head of a sequence (generic version):

$$\boxed{\begin{array}{l} \vdash [X] = \\ \quad \text{headOf} : \text{seq } X \rightarrow X \\ \hline \forall s : \text{seq } X \mid \#s > 0 \bullet \text{headOf}(s) = s(1) \end{array}}$$

This is defined only for non-empty sequences.

Example 6 : Generic Set Membership (Conceptual)

Set membership can be tested using the built-in 'in' operator in Z notation. In other contexts, you might define a membership test function, but Z notation provides this as a primitive predicate: x in S is true exactly when x is an element of S .

Example 7 : Generic Optional Type (Conceptual)

Optional values can be modeled with free types in Z notation. For a specific type, you would $\text{define} : Option ::= \text{none} \mid \text{some}\langle T \rangle$. Generic optional types require more complex setup beyond basic gedef blocks, as free types in Z are typically defined for specific types rather than being polymorphic.

Example 8 : Generic Swap Function

Swap the components of a pair:

$$\boxed{\begin{array}{l} \vdash [X, Y] = \\ \quad \text{swap} : X \times Y \rightarrow Y \times X \\ \hline \forall x : X \bullet \forall y : Y \bullet \text{swap}(x, y) = (y, x) \end{array}}$$

Converts (x, y) to (y, x) .

Example 9 : Generic Cardinality

Size of a set (for finite sets):

$$\boxed{\begin{array}{l} \vdash [X] = \\ \quad \text{size} : \mathbb{P} X \rightarrow \mathbb{N} \\ \hline \forall S : \mathbb{P} X \bullet \text{size}(S) = \#S \end{array}}$$

Returns the number of elements in a finite set.

Example 10 : Generic List Construction

Construct a singleton sequence:

$[X]$	<hr/>
$single : X \rightarrow \text{seq } X$	
$\forall x : X \bullet single(x) = \langle x \rangle$	

Wraps a value in a single-element sequence.

Example 11 : Using Generic Definitions

Once defined, generic functions can be instantiated for specific types:

$id_nat : \mathbb{N} \rightarrow \mathbb{N}$	<hr/>
$id_set : \mathcal{P} \mathbb{N} \rightarrow \mathcal{P} \mathbb{N}$	
$id_nat = identityGB[\mathbb{N}]$	
$id_set = identityGB[\mathcal{P} \mathbb{N}]$	

Here $identityGB[\mathbb{N}]$ is the $identityGB$ function instantiated for natural numbers, and $identityGB[\mathcal{P} \mathbb{N}]$ is instantiated for sets of natural numbers.

Example 12 : Generic Constants

Define a generic empty sequence:

$[X]$	<hr/>
$emptySeq : \text{seq } X$	
$emptySeq = \langle \rangle$	

This can be instantiated as $emptySeq[\mathbb{N}]$ for an empty sequence of naturals, $emptySeq[\text{Person}]$ for an empty sequence of persons, etc.