Generic Type Instantiation

Example 1 : Basic Type Instantiation

Apply type parameters to polymorphic types using square brackets:

seq $\mathbb{N}$
$\mathbb{P}\ X$
$\varnothing[\mathbb{Z}]$
These instantiate generic types for specific type arguments.

Example 2 : Empty Set Instantiation

$\quad$ $emptyNats : \mathbb{P}\ \mathbb{N}$
$\quad$ $emptyInts : \mathbb{P}\ \mathbb{Z}$
$\quad$ ────────────
$\quad$ $emptyNats = \varnothing[\mathbb{N}]$
$\quad$ $emptyInts = \varnothing[\mathbb{Z}]$

$\varnothing$ is generic; $\varnothing[N]$ is the empty set of naturals.

Example 3 : Sequence Type Instantiation

$\quad$ $numbers : \text{seq}\ \mathbb{N}$
$\quad$ $integers : \text{seq}\ \mathbb{Z}$
$\quad$ ────────────
$\quad$ $numbers = \langle 1, 2, 3 \rangle$
$\quad$ $integers = \langle -1, 0, 1 \rangle$

seq[N] is the type of sequences of natural numbers.

Example 4 : Power Set Instantiation

$\quad$ $subsetsOfN : \mathbb{P}(\mathbb{P}\ \mathbb{N})$
$\quad$ $example : \mathbb{P}(\mathbb{P}\ \mathbb{N})$
$\quad$ ────────────
$\quad$ $example = \{\{1, 2\}, \{3\}, \varnothing[\mathbb{N}]\}$

P(P N) is the power set of the power set of naturals (sets of sets).

Example 5 : Complex Type Parameters

$\quad$ $pairsSeq : \text{seq}(\mathbb{N} \times \mathbb{N})$
$\quad$ $pairsSet : \mathbb{P}(\mathbb{N} \times \mathbb{N})$
$\quad$ ────────────
$\quad$ $pairsSeq = \langle (1, 2), (3, 4) \rangle$
$\quad$ $pairsSet = \{(1, 2), (3, 4)\}$

Type parameters can be complex types like N cross N.

Example 6 : Nested Instantiation

$$seqOfSets : \text{seq}(\mathbb{P}\,\mathbb{N})$$
$$setOfSeqs : \mathbb{P}(\text{seq}\,\mathbb{N})$$

$$seqOfSets = \langle\{1,2\},\{3,4,5\}\rangle$$
$$setOfSeqs = \{\langle 1,2\rangle,\langle 3,4\rangle\}$$

seq[P N] is a sequence of sets. P[seq[N]] is a set of sequences.

Example 7 : Generic Function Instantiation

$[X]$

$$identity : X \longrightarrow X$$

$$\forall\, x : X \bullet identity(x) = x$$

$$id\_nat : \mathbb{N} \longrightarrow \mathbb{N}$$
$$id\_int : \mathbb{Z} \longrightarrow \mathbb{Z}$$

$$id\_nat = identity[\mathbb{N}]$$
$$id\_int = identity[\mathbb{Z}]$$

identity[N] instantiates the generic identity function for natural numbers.

Example 8 : Multiple Type Parameters

$[X, Y]$

$$pairType : X \times Y$$

$$natIntPair : \mathbb{N} \times \mathbb{Z}$$
$$intNatPair : \mathbb{Z} \times \mathbb{N}$$

$$natIntPair = pairType[\mathbb{N}, \mathbb{Z}]$$
$$intNatPair = pairType[\mathbb{Z}, \mathbb{N}]$$

Instantiate with multiple type arguments.

Example 9 : Bag Instantiation

$$natBag : \text{bag}\,\mathbb{N}$$
$$intBag : \text{bag}\,\mathbb{Z}$$

$$natBag = [\![1,1,2,3]\!]$$
$$intBag = [\![-1,0,1]\!]$$

bag[N] is the type of bags (multisets) of natural numbers.

Example 10 : Generic Relation Types

$$natRelation : \mathbb{N} \leftrightarrow \mathbb{N}$$
$$mixedRelation : \mathbb{N} \leftrightarrow \mathbb{Z}$$

$$natRelation = \{1 \mapsto 2, 2 \mapsto 4\}$$
$$mixedRelation = \{1 \mapsto -1, 2 \mapsto -2\}$$

Relations between different types.

Example 11 : Sequence of Pairs

$coordSeq : \text{seq}(\mathbb{N} \times \mathbb{N})$

$coordSeq = \langle (0, 0), (1, 1), (2, 4), (3, 9) \rangle$

A sequence where each element is a pair (coordinate).

Example 12 : Set of Functions

$square : \mathbb{N} \longrightarrow \mathbb{N}$
$double : \mathbb{N} \longrightarrow \mathbb{N}$
$successor : \mathbb{N} \longrightarrow \mathbb{N}$

$\forall\, n : \mathbb{N} \bullet square(n) = n * n$
$\forall\, n : \mathbb{N} \bullet double(n) = 2 * n$
$\forall\, n : \mathbb{N} \bullet successor(n) = n + 1$

$funcSet : \mathbb{P}(\mathbb{N} \longrightarrow \mathbb{N})$

$funcSet = \{ square, double, successor \}$

A set containing three functions.

Example 13 : Practical Example - Generic Pair

$[T]$

$makePair : T \times T \longrightarrow T \times T$

$\forall\, x, y : T \bullet makePair(x, y) = (x, y)$

Generic pair function that works with any type T.

Example 14 : Generic Pair Extraction

$[X]$

$fst : X \times X \longrightarrow X$
$snd : X \times X \longrightarrow X$

$\forall\, x, y : X \bullet fst(x, y) = x$
$\forall\, x, y : X \bullet snd(x, y) = y$

Generic first and second functions that work with any type X.

Example 15 : List of Lists

$matrix : \text{seq}(\text{seq}\,\mathbb{N})$

$matrix = \langle \langle 1, 2, 3 \rangle, \langle 4, 5, 6 \rangle, \langle 7, 8, 9 \rangle \rangle$

Nested sequences represent a 3x3 matrix.

Example 16 : Homogeneous Collections

Type instantiation ensures homogeneity:

$$\begin{array}{|l}
validSeq : \operatorname{seq} \mathbb{N} \\
\hline
validSeq = \langle 1, 2, 3 \rangle
\end{array}$$

$INVALID : seq[N] cannot contain non-naturals$

invalidSeq = $\langle 1, -1, 2 \rangle$ – would be type error if -1 not in N

Example 17 : Generic Queue

$$\begin{array}{l}
[T] \\
\hline
wrap : T \longrightarrow \operatorname{seq} T \\
unwrap : \operatorname{seq} T \longrightarrow T \\
\hline
\forall\, x : T \bullet wrap(x) = \langle x \rangle \\
\forall\, s : \operatorname{seq} T \bullet \#s > 0 \Rightarrow unwrap(s) = s(1)
\end{array}$$

Generic wrap/unwrap functions for working with sequences.

Example 18 : Type Synonyms with Instantiation

Type abbreviations can use instantiated types:

IntSet == P Z (set of integers)

NatSeq == seq N (sequence of naturals)

Coordinate == N cross N (pair of naturals)

Path == seq Coordinate (sequence of coordinates)

These provide readable names for complex instantiated types.

Example 19 : Higher - Order Types

$$\begin{array}{|l}
transformers : \mathbb{P}(\mathbb{N} \longrightarrow \mathbb{N}) \\
sequences : \mathbb{P}(\operatorname{seq} \mathbb{N}) \\
\hline
transformers = \{square, double\} \\
sequences = \{\langle 1, 2 \rangle, \langle 3, 4 \rangle\}
\end{array}$$

Sets of functions and sequences.

Example 20 : Instantiation elem Quantifiers

$\forall\, s : \operatorname{seq} \mathbb{N} \bullet \#s \geq 0$
$\exists\, p : \mathbb{N} \times \mathbb{N} \bullet p.1 = p.2$
Quantifying over instantiated generic types.

Example 21 : Partial Function Types

$$lookup : \mathbb{N} \nrightarrow seq\,\mathbb{N}$$
$$mapping : \mathbb{N} \leftrightarrow \mathbb{N}$$

$$lookup = \{1 \mapsto \langle 10, 20 \rangle, 2 \mapsto \langle 30, 40 \rangle\}$$
$$mapping = \{1 \mapsto 2, 2 \mapsto 4, 3 \mapsto 6\}$$

Partial functions with instantiated types.

Example 22 : Best Practices

When using generic instantiation:

1. Be explicit about type parameters for clarity

2. Use consistent naming (seq[T], not mixed seq T and T seq)

3. Document what type parameters represent

4. Instantiate at the right abstraction level

5. Prefer generic definitions over duplicate specific definitions

6. Type-check that instantiations make sense

Example 23 : Common Patterns

Common instantiation patterns:

- seq[T] for sequences of T

- P[T] for sets of T

- $T \nrightarrow U$ for partial functions

- T $\leftrightarrow$ U for relations

- bag[T] for multisets of T

- T cross U for pairs

- $T \longrightarrow U$ for total functions

Example 24 : Type Safety

Generic instantiation provides type safety:

$$safe : seq\,\mathbb{N}$$

$$safe = \langle 1, 2, 3 \rangle$$

Type checker ensures all elements match the instantiated type.

COMPILE ERROR examples:

- seq[N] = ⟨1, 2, 'a'⟩ – 'a' is not a natural

- P[N] = $\{1, 2, -1\}$ – -1 might not be in N depending on N definition

- bag[Z] = [[1, "hello"]] – "hello" is not an integer

The type system catches these errors at compile/check time.