Generic Type Instantiation

Example 1 : Basic Type Instantiation

Apply type parameters to polymorphic types using square brackets:

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

Example 2 : Empty Set Instantiation

$$\begin{array}{|l}
\mathit{emptyNats} : \mathbb{P}\,\mathbb{N} \\
\mathit{emptyInts} : \mathbb{P}\,\mathbb{Z} \\
\hline
\mathit{emptyNats} = \varnothing[\mathbb{N}] \\
\mathit{emptyInts} = \varnothing[\mathbb{Z}]
\end{array}$$

∅ is generic; ∅[N] is the empty set of naturals.

Example 3 : Sequence Type Instantiation

$$\begin{array}{|l}
\mathit{numbers} : \mathrm{seq}\,\mathbb{N} \\
\mathit{integers} : \mathrm{seq}\,\mathbb{Z} \\
\hline
\mathit{numbers} = \langle 1, 2, 3 \rangle \\
\mathit{integers} = \langle -1, 0, 1 \rangle
\end{array}$$

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

Example 4 : Power Set Instantiation

$$\begin{array}{|l}
\mathit{subsetsOfN} : \mathbb{P}(\mathbb{P}\,\mathbb{N}) \\
\mathit{example} : \mathbb{P}(\mathbb{P}\,\mathbb{N}) \\
\hline
\mathit{example} = \{\{1, 2\}, \{3\}, \varnothing[\mathbb{N}]\}
\end{array}$$

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

Example 5 : Complex Type Parameters

$$\begin{array}{|l}
\mathit{pairsSeq} : \mathrm{seq}(\mathbb{N} \times \mathbb{N}) \\
\mathit{pairsSet} : \mathbb{P}(\mathbb{N} \times \mathbb{N}) \\
\hline
\mathit{pairsSeq} = \langle (1, 2), (3, 4) \rangle \\
\mathit{pairsSet} = \{(1, 2), (3, 4)\}
\end{array}$$

Type parameters can be complex types like N cross N.

Example 6 : Nested Instantiation

$$\begin{array}{|l}
seqOfSets : \mathrm{seq}(\mathbb{P}\,\mathbb{N}) \\
setOfSeqs : \mathbb{P}(\mathrm{seq}\,\mathbb{N}) \\
\hline
seqOfSets = \langle \{1, 2\}, \{3, 4, 5\} \rangle \\
setOfSeqs = \{ \langle 1, 2 \rangle, \langle 3, 4 \rangle \}
\end{array}$$

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

Example 7 : Generic Function Instantiation

$$\begin{array}{|l}
[X] \\
\hline
identity : X \longrightarrow X \\
\hline
\forall\, x : X \bullet identity(x) = x
\end{array}$$

$$\begin{array}{|l}
id_nat : \mathbb{N} \longrightarrow \mathbb{N} \\
id_int : \mathbb{Z} \longrightarrow \mathbb{Z} \\
\hline
id_nat = identity[\mathbb{N}] \\
id_int = identity[\mathbb{Z}]
\end{array}$$

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

Example 8 : Multiple Type Parameters

$$\begin{array}{|l}
[X, Y] \\
\hline
pairType : X \times Y
\end{array}$$

$$\begin{array}{|l}
natIntPair : \mathbb{N} \times \mathbb{Z} \\
intNatPair : \mathbb{Z} \times \mathbb{N} \\
\hline
natIntPair = pairType[\mathbb{N}, \mathbb{Z}] \\
intNatPair = pairType[\mathbb{Z}, \mathbb{N}]
\end{array}$$

Instantiate with multiple type arguments.

Example 9 : Bag Instantiation

$$\begin{array}{|l}
natBag : \mathrm{bag}\,\mathbb{N} \\
intBag : \mathrm{bag}\,\mathbb{Z} \\
\hline
natBag = [\![1, 1, 2, 3]\!] \\
intBag = [\![-1, 0, 1]\!]
\end{array}$$

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

Example 10 : Generic Relation Types

$$\begin{array}{|l}
natRelation : \mathbb{N} \leftrightarrow \mathbb{N} \\
mixedRelation : \mathbb{N} \leftrightarrow \mathbb{Z} \\
\hline
natRelation = \{1 \mapsto 2, 2 \mapsto 4\} \\
mixedRelation = \{1 \mapsto -1, 2 \mapsto -2\}
\end{array}$$

Relations between different types.

Example 11 : Sequence of Pairs

$$\begin{array}{|l}
coordSeq : \mathrm{seq}(\mathbb{N} \times \mathbb{N}) \\
\hline
coordSeq = \langle (0,0), (1,1), (2,4), (3,9) \rangle
\end{array}$$

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

Example 12 : Set of Functions

$$\begin{array}{|l}
square : \mathbb{N} \longrightarrow \mathbb{N} \\
double : \mathbb{N} \longrightarrow \mathbb{N} \\
successor : \mathbb{N} \longrightarrow \mathbb{N} \\
\hline
\forall n : \mathbb{N} \bullet square(n) = n * n \\
\forall n : \mathbb{N} \bullet double(n) = 2 * n \\
\forall n : \mathbb{N} \bullet successor(n) = n + 1
\end{array}$$

$$\begin{array}{|l}
funcSet : \mathbb{P}(\mathbb{N} \longrightarrow \mathbb{N}) \\
\hline
funcSet = \{square, double, successor\}
\end{array}$$

A set containing three functions.

Example 13 : Practical Example - Generic Pair

$$\begin{array}{|l}
[T] \\
\hline
makePair : T \times T \longrightarrow T \times T \\
\hline
\forall x, y : T \bullet makePair(x, y) = (x, y) \\
\hline
\end{array}$$

Generic pair function that works with any type T.

Example 14 : Generic Pair Extraction

$$\begin{array}{|l}
[X] \\
\hline
fst : X \times X \longrightarrow X \\
snd : X \times X \longrightarrow X \\
\hline
\forall x, y : X \bullet fst(x, y) = x \\
\forall x, y : X \bullet snd(x, y) = y \\
\hline
\end{array}$$

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

Example 15 : List of Lists

$$\begin{array}{|l}
matrix : \mathrm{seq}(\mathrm{seq}\,\mathbb{N}) \\
\hline
matrix = \langle \langle 1, 2, 3 \rangle, \langle 4, 5, 6 \rangle, \langle 7, 8, 9 \rangle \rangle
\end{array}$$

Nested sequences represent a 3x3 matrix.

Example 16 : Homogeneous Collections

Type instantiation ensures homogeneity:

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

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

invalidSeq = ⟨1, -1, 2⟩ – would be type error if -1 not in N

Example 17 : Generic Queue

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

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

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

Sets of functions and sequences.

Example 20 : Instantiation elem Quantifiers

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

Example 21 : Partial Function Types

$$
\begin{array}{|l}
lookup : \mathbb{N} \pfun \operatorname{seq} \mathbb{N} \\
mapping : \mathbb{N} \leftrightarrow \mathbb{N} \\
\hline
lookup = \{1 \mapsto \langle 10, 20 \rangle, 2 \mapsto \langle 30, 40 \rangle\} \\
mapping = \{1 \mapsto 2, 2 \mapsto 4, 3 \mapsto 6\}
\end{array}
$$

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

- $T \pfun U$ 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:

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

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.