Phase 34 : Finite Partial Functions

Finite partial functions are partial functions with a finite domain. The *operator* $⇻$ *denotes* a finite partial function, meaning the function may not be defined everywhere, and its domain must be finite.

Example 1 : Basic Finite Function Type

A finite partial function from X to Y is a partial function whose domain is finite. This is useful when modeling resources with limited capacity or databases with a bounded number of entries.

$$[\mathit{Year}, \mathit{TableFF}]$$

$$\begin{array}{|l}
\mathit{records} : \mathit{Year} ⇻ \mathit{TableFF} \\
\hline
\#(\operatorname{dom} \mathit{records}) \leq 1000
\end{array}$$

The records relation is a finite partial function because:

1. It is partial: not all years need to have a table

2. Its domain is finite: at most 1000 years have records

3. It is functional: each year maps to at most one table

Example 2 : Comparison with Other Function Types

Comparison of function type operators:

- Total function ($\longrightarrow$): defined for all inputs, domain is entire source set

- Partial function ($⇸$): may not be defined for all inputs, domain may be infinite

- Finite partial function ($⇻$): partial function with finite domain

- Total bijection ($⤖$): defined everywhere, injective and surjective

$$[X, Y]$$

$$\begin{array}{|l}
\mathit{totalFunc} : X \longrightarrow Y \\
\mathit{partialFunc} : X ⇸ Y \\
\mathit{finitePartialFunc} : X ⇻ Y \\
\mathit{totalBij} : X ⤖ Y \\
\hline
\operatorname{dom} \mathit{totalFunc} = X \land \operatorname{dom} \mathit{totalBij} = X \land (\forall x1, x2 : X \mid \mathit{totalBij}(x1) = \mathit{totalBij}(x2) \bullet x1 = x2) \land \operatorname{ran} \mathit{totalBij} =
\end{array}$$

Example 3 : Finite Functions elem Practice

Real-world example: A database table storing customer preferences.

$$[CustomerID, Preference]$$

$$\begin{array}{|l} preferences : CustomerID \mathrel{+\!\!\!\!\twoheadrightarrow} Preference \\ \hline \#(\operatorname{dom} preferences) \leq 10000 \end{array}$$

This models a preference database where:

- Not all customers have preferences recorded (partial)

- The database has a capacity limit (finite domain)

- Each customer maps to one preference value (functional)

Example 4 : Operations on Finite Functions

$$\begin{array}{|l} f : \mathbb{N} \mathrel{+\!\!\!\!\twoheadrightarrow} \mathbb{N} \\ g : \mathbb{N} \mathrel{+\!\!\!\!\twoheadrightarrow} \mathbb{N} \\ \hline f = \{1 \mapsto 10, 2 \mapsto 20, 3 \mapsto 30\} \land \\ \quad g = \{10 \mapsto 100, 20 \mapsto 200\} \land \\ \quad \#(\operatorname{dom} f) = 3 \land \\ \quad \#(\operatorname{dom} g) = 2 \end{array}$$

We can compose finite functions, but the result may not be finite unless proven.

$$\begin{array}{|l} f\_composed : \mathbb{N} \mathrel{+\!\!\!\!\twoheadrightarrow} \mathbb{N} \\ \hline f\_composed = g \circ f \land \#(\operatorname{dom} f\_composed) \leq \#(\operatorname{dom} f) \end{array}$$

The composition has a domain size bounded by the domain of f, since we can only compose where f's range intersects g's domain.

Example 5 : Domain Restrictions land Finiteness

$$[Title, Length, ViewDate]$$

$$\begin{array}{|l} viewed : Title \mathrel{+\!\!\!\!\twoheadrightarrow} ViewDate \\ recent\_viewed : Title \mathrel{+\!\!\!\!\twoheadrightarrow} ViewDate \\ recentSet : \mathbb{P}\ Title \\ \hline recentSet = \{\ t : Title \mid t \in \operatorname{dom} viewed\ \} \land \\ \quad recent\_viewed = recentSet \lhd viewed \land \\ \quad \#(\operatorname{dom} recent\_viewed) \leq \#(\operatorname{dom} viewed) \end{array}$$

Restricting a finite partial function preserves finiteness. The recent_viewed function has a domain that is a subset of viewed's domain, so it remains finite.