Composition Pipelines

Example 1 : Simple Three - Stage Pipeline

A three-stage numeric processing pipeline.

$$
\begin{array}{|l}
addOne : \mathbb{N} \longrightarrow \mathbb{N} \\
doubleCP : \mathbb{N} \longrightarrow \mathbb{N} \\
squareCP : \mathbb{N} \longrightarrow \mathbb{N} \\
pipeline : \mathbb{N} \longrightarrow \mathbb{N} \\
\hline
\forall\, n : \mathbb{N} \bullet addOne(n) = n + 1 \\
\forall\, n : \mathbb{N} \bullet doubleCP(n) = 2 * n \\
\forall\, n : \mathbb{N} \bullet squareCP(n) = n * n \\
pipeline = squareCP \fatsemi doubleCP \fatsemi addOne
\end{array}
$$

pipeline(3) = squareCP(doubleCP(addOne(3))) = squareCP(doubleCP(4)) = squareCP(8) = 64

Example 2 : String Pipeline

$[CharCP]$

$$
\begin{array}{|l}
str1 : \text{seq}\ CharCP \longrightarrow \text{seq}\ CharCP \\
str2 : \text{seq}\ CharCP \longrightarrow \text{seq}\ CharCP \\
str3 : \text{seq}\ CharCP \longrightarrow \text{seq}\ CharCP \\
stringPipeline : \text{seq}\ CharCP \longrightarrow \text{seq}\ CharCP \\
\hline
stringPipeline = str3 \fatsemi str2 \fatsemi str1
\end{array}
$$

Demonstrates composition of string-processing functions.

Example 3 : Best Practices

When building pipelines:

1. Break complex operations into simple stages
2. Each stage should have a single responsibility
3. Test individual functions before composing
4. Use meaningful names for pipeline stages
5. Document the flow direction (comp applies right-to-left)