Syntax Environment Examples

$[NAME]$

Example 1 : Simple Enumeration

For simple free types with several branches, syntax provides

column-aligned layout:

$$OP1 ::= plus \mid minus \mid times \mid divide$$

This generates aligned columns with the ::= separator clearly visible.

Example 2 : Parameterized Constructors

Constructors can have type parameters using angle brackets:

$$Tree ::= leaf\langle\!\langle \mathbb{N} \rangle\!\rangle \mid branch\langle\!\langle Tree \times Tree \rangle\!\rangle$$

Parameters are rendered with special data delimiters in LaTeX.

Example 3 : Multiple Definitions with Grouping

Blank lines between definitions create visual groups using \also:

$$Status2 ::= active2 \mid inactive2 \mid pending2$$
$$Response ::= ok \mid error\langle\!\langle \mathbb{N} \rangle\!\rangle \mid timeout$$

The blank line causes the also command to be generated, creating vertical

separation between the two type definitions.

Example 4 : Complex BNF - Style Definitions

Syntax environment is ideal for BNF-style grammar definitions:

$$Expr ::= const\langle\!\langle \mathbb{N} \rangle\!\rangle \mid var\langle\!\langle NAME \rangle\!\rangle \mid binop\langle\!\langle OP1 \times Expr \times Expr \rangle\!\rangle$$
$$Stmt ::= assign\langle\!\langle NAME \times Expr \rangle\!\rangle \mid seq\langle\!\langle Stmt \times Stmt \rangle\!\rangle \mid skip$$

This creates a professionally formatted grammar specification.

Comparison : Syntax vs Zed Blocks

Standard zed blocks work for simple types:

$$Color1 ::= red1 \mid green1 \mid blue1$$

But syntax environment provides better alignment for complex types:

$$Color2 ::= red2 \mid green2 \mid blue2$$
$$Shape ::= circle\langle\!\langle \mathbb{N} \rangle\!\rangle \mid rectangle\langle\!\langle \mathbb{N} \times \mathbb{N} \rangle\!\rangle \mid polygon\langle\!\langle \mathrm{seq}\ \mathbb{N} \rangle\!\rangle$$

The column alignment makes the structure more readable.