Using ontologies allows us to statically analyze Ptolemy models in a principled way. The first step to doing so is to represent the domain of interest as a lattice-based ontology.

**Example Ontologies.**

Infinite Ontologies

While many types of knowledge can be encoded in finite ontologies, infinite ontologies can express a larger class of properties. We have found two patterns of infinite ontologies to be broadly useful:

- **Infinite Flat Lattices:** Useful for including value information into the ontologies.
- **Recursive Lattices:** Useful for including structured information corresponding to structured data types.

**Infinite Flat Lattice Pattern**

In the infinite flat lattice pattern, we allow lattices that are infinite in one dimension at specific points in the lattice. This is most often useful as a way of parameterizing a concept by a value in order to make something resembling a dependent type system in which values are incorporated into types.

**Example: Constant Propagation**

**Infinite Recursive Lattice Pattern**

Infinite Recursive Lattices may be contained within a concept in the lattice. In the lattice to the right, for example, the rightmost concept is isomorphic to the entire lattice.

A classic example of a recursive lattice is a type lattice that contains array or record types. Since arrays (or records) may recursively contain arrays or records within them, this creates an infinite lattice of potential types.

**Example: Monotonicity Analysis**

In addition to dataflow-based graphs, Ptolemy expressions can also be analyzed with our solver. This example shows an analysis that determines whether expressions are monotonic, antimonotonic, constant, or not.

Monotonic:

\[ x \leq y \implies f(x) \leq f(y) \]

Antimonic:

\[ x \leq y \implies f(x) \geq f(y) \]

Constant:

\[ f(x) = f(y) \]