Discrete Lattice Functions

by Sven Nilsen, 2020

In this paper I formalize discrete lattice functions, which have a boolean and a monotonic form.

Assume that you have a function from natural numbers to booleans:

f : nat → bool

A discrete lattice[1] function of `f` is a function `g` defined as following:

g(n) = ∑ x [0, n) { if f(x) {1} else {0} }

Discrete lattice functions are monotonic[2], but contains the same information as the original function.
Therefore, one can think about `f` as a boolean representation of a discrete lattice function.
The representation `g` can be thought of as a monotonic representation of a discrete lattice function.

Any function of the type `nat → bool` is a discrete lattice function, so why not use this definition?
The problem is emphasize the context in which a function is talked about: Boolean vs monotonic.
The word “discrete” refers to the order of natural numbers, not discrete lattices in general.
The full description is needed for an unambiguous meaning: “Discrete Lattice Function”.

Like any lattice[1], there is a supremum and an infimum for both representations:

\true	Supremum in boolean representation
\false	Infimum in boolean representation
id	Supremum in monotonic representation
\0	Infimum in monotonic representation

Any monotonic function[2] with the following property is “above `id`”:

g(x) > x

When a monotonic function is above `id`, it is not a discrete lattice function.
However there exists functions below `id` which are not discrete lattice functions.

For example:

g(x) = if x < 2 { 0 } else { 2 }

This is impossible to express in the boolean representation.

Discrete lattice functions naturally model ordered counting.
There are some restrictions to this kind of counting that does hold for functions or sums in general.
The counting can only increase with `1` at maximum for each step.

References:

[1] “Lattice (order)”
Wikipedia
https://en.wikipedia.org/wiki/Lattice_(order)

[2] “Monotonic function”
Wikipedia
https://en.wikipedia.org/wiki/Monotonic_function