Information Optimal Functions

by Sven Nilsen, 2018

An information optimal function has required number of input bits just enough to decide its output:

information_optimal := \(f) = ceil($\log_2$(|∀f|)) == ceil($\log_2$(|∃f|))

In Boolean algebra the functions `not : bool → bool` and `id : bool → bool` are examples of information optimal functions. The functions `$\text{false}_1$ : bool → bool` and `$\text{true}_1$ : bool → bool` are not information optimal because they always return a single value which requires 0 bits, while the input requires 1 bit.

The largest difference between input and output for deterministic functions is given by a sequence:

$2^N - 1$

For example, 16 inputs requires `N=4` bits, which means `$2^4 - 1 = 7$` values are allowed to be lost while still keeping the function information optimal. This property is preserved under composition.

An information optimal function is almost structure preserving. All functions that are structure preserving have the following property:

|∀f| == |∃f|
$\log_2$(|∀f|) == $\log_2$(|∃f|)

The difference is that information optimal functions uses the `ceil` function before equality check. Therefore, structure preserving functions is a subset of information optimal functions.

Information optimal functions are used to reason about whether input and output types are efficiently compressed. However, there are many useful functions that are not information optimal, such as AND and OR. Such functions are useful because they decide by combining information from their arguments. One can not create information optimal functions by eliminating some of the arguments:

and[unit × id → id] <=> {}
and[id × unit → id] <=> {}
or[unit × id → id] <=> {}
or[id × unit → id] <=> {}

On the other hand, if e.g. `not` is extended with an extra argument that is ignored, then it is possible to get a normal path by eliminating one argument:

$\text{not}_{\text{ignore\_fst}}$[unit × id] <=> not
$\text{not}_{\text{ignore\_snd}}$[id × unit] <=> not

Information optimal functions are indirectly useful because one can analyze different methods of detecting inefficient encodings of data, by extending known functions that are information optimal.